НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

На правах рукопису
УДК 004.056.53

До захисту допущено
В. о. завідувача кафедри ММСА
О.Л.Тимощук
«___» _____ 2020 р

Магістерська дисертація

на здобуття ступеня магістра за спеціальністю 122 Комп’ютерні науки
на тему: «Виявлення мережевих аномалій за допомогою систем штучного інтелекту»

Виконав
студент II курсу, групи КА-93 мп
Хархонов Антон Володимирович _____

Керівник:
доцент кафедри ММСА, к.т.н, доц.
Дідковська М.В. _____

Рецензент:
доцент кафедри програмного забезпечення комп'ютерних систем
КПІ ім. Ігоря Сікорського, к.т.н., доц. Заболотня Т.М. _____

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань

Студент _____

КИЇВ

2020

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

Рівень вищої освіти — другий (магістерський)

Спеціальність — 122 «Комп'ютерні науки»

ЗАТВЕРДЖУЮ
В. о. завідувача кафедри ММСА
О. Л. Тимощук
«___»_____ 2020 р.

ЗАВДАННЯ

на магістерську дисертацію студента Хархонова Антона Володимировича

1. Тема дисертації: «Виявлення мережевих аномалій за допомогою систем штучного інтелекту», науковий керівник дисертації Дідковська Марина Віталієвна, к.ф.-м.н., доцент, затверджені наказом по університету від « 02 » листопада 2020 р. № 3182-с

2. Термін подання студентом дисертації: 16.12.2020

3. Об'єкт дослідження: Мережеві аномалії

4. Предмет дослідження**:** Використання мережевих аномалій для виявлення вторгнень

5. Перелік завдань, які потрібно розробити:

1) дослідити сучасний стан та особливості застосування методів машинного навчання у вирішенні проблеми виявлення мережевих аномалій

2) побудова моделей систем виявлення мережевих аномалій на основі досліджених алгоритмів та методів машинного навчання

3) проаналізувати отримані результати побудованих моделей

5) розробити стартап-проект виведення на ринок результатів дослідження

6) розробити концептуальні висновки за результатами наукового дослідження

6. Орієнтований перелік графічного (ілюстративного) матеріалу:

1) діаграми архітектур

2) результати роботи програмного продукту

3) таблиці у розділі стартап-проекту

7. Дата видачі завдання: 01 вересня 2020р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації
1.	Концептуальний вступ дисертації. Формулювання об’єкта, предмета, цілі, завдань, новизни, практичної значущості результатів	01.09.2020—10.09.2020
2.	Перший розділ. Загальний огляд предметної області	11.09. 2020—28.09. 2020
3.	Другий розділ. Аналіз задачі виявлення мережевих аномалій	28.09. 2020—21.10. 2020
4.	Третій розділ. Побудова моделей	21.10. 2020—15.11. 2020
5.	Четвертий розділ. Стартап-проект	15.11. 2020—20.11. 2020
6.	Концептуальні висновки. Перспективи розвитку отриманих рішень	20.11. 2020—25.11. 2020

Студент Хархонов А.В.

Науковий керівник дисертації Дідковська М.В.

РЕФЕРАТ

Магістерська дисертація: 123с., 4 ч., 35 табл., 16 рис., 1 дод., 48 джерел.

ШТУЧНИЙ ІНТЕЛЕКТ, МАШИННЕ НАВЧАННЯ, КІБЕРБЕЗПЕКА, ВИЯВЛЕННЯ МЕРЕЖЕВИХ АНОМАЛІЙ, PYTHON, CICIDS2017

Об’єктом дослідження є мережеві аномалії.

Предметом дослідження є використання мережевих аномалій для виявлення вторгнень.

Мета роботи – розробити систему виявлення мережевих аномалій на основі досліджених алгоритмів та методів машинного навчання.

Методи дослідження – статистичні методи, класифікаційні метод, методи на базі кластеризації, методи на базі знань, комбіновані методи.

Актуальність – виявлення вторгнень та миттєве сповіщення адміністраторів мережі про потенційну загрозу інфраструктурі. Система перешкоджає зловмисникам отримати несакціонований доступ до мережі за допомогою як відомих так і невідомих атак.

Новизна – на відміну від ручного адміністрування, автоматизована система дозволяє зекономити ресурси та не допускає помилки через людський фактор.

Результати дослідження – побудована модель для автоматичного виявлення мережевих аномалій для запобігання вторгнень у мережу або інфраструктуру.

ABSTRACT

Masters' thesis: 123p., 4 s., 35 tabl., 16 fig., 1 appendix., 48 references.

ARTIFICIAL INTELLIGENCE, MACHINE LEARNING, NETWORK SECURITY, NETWORK ANOMALY DETECTION, PYTHON, CICIDS2017

The object of this research is network anomalies.

The subject of the research is the use of network anomalies for intrusion detection.

The purpose of the work is to develop a system for detecting network anomalies based on the studied algorithms and methods of machine learning.

Methods of the study – statistical methods, classification, clustering, knowledge base, combination learning.

The relevance of the study – Intrusion detection and immediate notification of network administrators about a potential threat to the infrastructure. The system prevents intruders from accessing the network through both known and unknown attacks.

Novelty – In contrast to manual administration, an automated system saves resources and does not make mistakes due to human factor.

The results of the study – A model was built to automatically detect network anomalies to prevent intrusion into the network or infrastructure.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

U2R	– User to Root
R2L	– Remote to Local
IDS	– Intrusion detection system
HIDS	– Host-based IDS
NIDS	– Network-based IDS
ANIDS	– Anomaly-based network intrusion detection system
DDoS	– Distributed denial of service
IOF	– Inverse Occurrence Frequency
OF	– Occurrence Frequency
HIDE	– Hierarchical Network Intrusion Detection
IDA	– Intrusion Detection Agents
LEAD	– Learning rules for anomaly detection
HTTP	– HyperText Transfer Protocol
PAYL	– Payload-based Network Intrusion Detection
FSAS	– Flow-based Statistical Aggregation Scheme
ADAM	– Automated Data Analysis and Mining
DGSOT	– Dynamically Growing Self-Organizing Tree
MINDS	– Minnesota Intrusion Detection System
GBID	– Genetic-based Intrusion Detection
RT-UNNID	– Realtime Unsupervised Neural Network Intrusion Detection
TCP	– Transmission Control Protocol
UDP	– User Datagram Protocol
ICMP	– Internet Control Message Protocol
FIRE	– Fuzzy Intrusion Recognition Engine

STAT – State Transition Analysis Tool

SSO – Site Security Officer

HMMPayl – Hidden Markov Models Payload

IPS – Intrusion Prevention System

FLIPS – Feedback Learning IPS

ISR – Instruction Set Randomization

QDA – Quadratic discriminant analysis

ID3 – Iterative Dichotomiser 3

MLP – Multilayer perceptron

KNN – Quadratic discriminant analysis

ВСТУП

Щодня мільйони людей і сотні тисяч установ спілкуються один з одним через Інтернет. За останні два десятиліття кількість людей, що користуються Інтернетом, зросла дуже швидко. Паралельно з цим число атак в інтернеті зростає з кожним днем.

Забезпечення інформаційної безпеки є першочерговим завданням кожної держави. В епоху комп'ютеризації і автоматизації проблема комп'ютерної безпеки виходить на перший план. Одним із завдань, яке доводиться вирішувати в контексті інформаційної безпеки є захист інформації, яка зберігається, обробляється і передається в комп'ютерних системах і мережах. Однією із загроз комп'ютерної безпеки є мережеві атаки. Під мережевою (або хакерською) атакою розуміється інформаційний руйнівний вплив, який здійснюється програмним методом і спрямований на розподілену обчислювальну систему. У залежності від методу організації мережевої атаки і засобів, які використовуються виділяють кілька різновидів мережевих атак - DDoS, U2R, R2L і Probe. Існують два напрямки забезпечення комп'ютерної інформаційної безпеки. Першим напрямом є запровадження адміністративних та кримінальних покарань за вчинення комп'ютерних злочинів. Другим напрямом є розробка апаратно-програмних засобів виявлення і захисту від мережевих вторгнень і шкідливих програм.

Незважаючи на те, що для запобігання таких атак використовуються сигнатурні методи, вони є невдалими в боротьбі з атаками нульового дня. З іншого боку, підхід, заснований на аномаліях, є альтернативним рішенням для мережевих атак і має здатність виявляти також атаки нульового дня.

Методи, засновані на сигнатурних підписах, використовують створену ними базу даних для виявлення атак. Цей метод досить успішний, але бази даних потребують постійного оновлення та обробці нової інформації про атаки. Більш того, навіть якщо бази даних оновлюються, вони уразливі для атак нульового дня (раніше непомічених). Оскільки таких атак немає в базі даних, вони не можуть їм запобігти. Підхід, заснований на аномаліях, фокусується на виявленні незвичайного мережевої поведінки шляхом вивчення мережевого потоку. Цей метод, який був успішним у виявленні атак, з якими він не стикався раніше, так само ефективний проти атак "нульового дня".

Крім того, більше половини підключень до інтернету шифрується за допомогою протоколів SSL / TLS (Secure Sockets Layer / Transport Layer Security), і цей показник зростає з кожним днем. Через неможливість спостереження за вмістом зашифрованого інтернет-потоку, методи, засновані на сигнатурах, не працюють ефективно з цим типом даних, однак аномальний підхід аналізує дані по їх загальним властивостям, таким як розмір, час з'єднання, і кількість пакетів. Таким чином, йому не потрібно бачити вміст повідомлення, і він також може проводити аналіз зашифрованих протоколів. Завдяки всім цим перевагам, метод виявлення аномалій інтенсивно використовується для виявлення і попередження мережевих атак.

РОЗДІЛ 1 ЗАГАЛЬНИЙ ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ

На сьогодні, із збільшенням інтернет користувачів, такі мережі стають дедалі частіше ціллю хакерських атак. Спроба або загроза вторгнення визначаються як навмисна і несанкціонована спроба отримати доступ до інформації, маніпулювання інформацією або спроби зробити систему ненадійною або непридатною для використання. По мірі неухильного розвитку мережевих комп'ютерних систем та сучасних технологій, росте потреба в системах, здатних виявляти мережеві вторгнення, які представляють собою величезний ризик для безпеки. Як приклад, можна привести широкомасштабну атаку на відмову в обслуговуванні (DDoS атака) на компанію Google в 2017 році. Атака була зроблена за допомогою ботнета інтернет речей, який представляє собою мережу приватних комп'ютерів, заражених шкідливим програмним забезпеченням і контрольованих як група без відома власників.

Вторгнення – це сукупність дій, спрямованих на компрометацію безпеки комп'ютерних та мережевих компонентів, порушення конфіденційності, цілісності та доступності [1]. Це може бути зроблено внутрішнім або зовнішнім агентом для отримання несанкціонованого проникнення і контролю над механізмами безпеки. Для захисту інфраструктури мережевих систем, системи виявлення вторгнень (IDS) забезпечують добре налагоджені механізми, які збирають та аналізують інформацію із різних областей всередині хоста або мережі для виявлення можливих порушень безпеки. Такі системи включають:

а) функції моніторингу та аналізу користувача, системи та мережевих активностей;
б) систему налаштувань для генерації звіт про можливі вразливості;
в) оцінку цілісності системи та файлів;

г) розпізнавання закономірностей типових атак;

д) аналіз аномальної активності;

е) відстеження порушень користувацької політики.

IDS використовує оцінку вразливості для оцінки безпеки хоста або мережі. Виявлення вторгнення працює виходячи з припущення, що діяльність взлому системи помітно відрізняється від звичайної діяльності системи і, таким чином, піддається виявленню.

1.1 Огляд можливих атак на комп’ютери та мережі

Атаки розділяють на два види: зовнішні та внутрішні [2]. Зовнішні атаки є неавторизованими користувачами машин, які вони атакують, в той час як внутрішні порушники мають дозвіл на доступ до системи але не мають привелегій для супер користувацького режиму. Замаскований внутрішній порушник входить в систему як і інші користувачі, які мають законний доступ до конфіденційних даних, а таємний внутрішній зловмисник, найнебезпечніший, має можливість відключити аудитерський контроль для себе таким чином, що навіть після виявлення вторгнення, в логах системи не буде помітно пересування зловмисника по системі. Такі атаки можуть бути комбінованими або гібридними [3], в залежності від складності проникнення в систему та наявності доступу до головних модулів або компонентів мережі [4]. Перелік основних типів атак наведено в таблиці 1.1.

Таблиця 1.1 – Види комп’ютерних атак: характеристики та приклади

Назва атаки	Характеристика	Приклад
Вірус	Самовідтворювана програма, яка заражає систему без відома і дозволу користувача. Підвищує рівень зараження мережевої файлової системи, якщо доступ до системи здійснюється з іншого комп’ютера.	Trivial.88.D, Polyboot.B, Tuager
Хробак	Програма розповсюджується через мережеві служби в комп’ютерних системах без втручання користувача. Може завдати серйозної шкоди мережі, споживаючи пропускну здатність мережі	SQL Slammer, Mydoom, CodeRed Nimba
Троян	Шкідлива програма, яка не може самовідтворюватись, але може викликати серйозні проблеми з безпекою в комп'ютерній системі. Встановлюється як корисна програма, але насправді в неї секретний код, який може несанкціонований доступ до системи, дозволяючи програмі робити що завгодно в системі, та може бути викликаний коли завгодно, так як хакер отримує контроль над системою без дозволу користувача	HookDump, Back Orifice, Pinch, TDL-4, Trojan.Winlock

Продовження таблиці 1.1

Назва атаки	Характеристика	Приклад
DoS-атака	Комплекс дій, націлених на блокування доступу до системних або мережевих ресурсів. Це реалізується шляхом примусу цільвого комп'ютера (комп'ютерів) до перезавантаження або споживання ресурсів. Користувачі цих систем не можуть адекватно працювати через відсутність обслуговування або перешкод, що створюються нестачею обчислювальних ресурсів	Buffer overflow, ping of death (PoD), TCP SYN, smurf, teardoop
Мережеві атаки	Будь-який процес, який використовується для зловмисних спроб підірвати безпеку мережі, починаючи з канального і закінчуючи прикладним рівнем, за допомогою різних засобів, такий як маніпуляції з мережевими протоколами. Незаконне використання облікових записів і прав користувачів, виконання дій з видалення мережевих ресурсів та пропускної здатності, виконання дій, що перешкоджають доступу авторизованих користувачів до мережевих служб та ресурсів.	Packet injection, SYN flood

Продовження таблиці 1.1

Назва атаки	Характеристика	Приклад
Фізичні атаки	Спроби пошкодити фізичні компоненти мережі або комп’ютера	Cool boot, evil maid
Злом пароля	Ціль – отримати пароль користувача в короткий період часу, зазвичай на таку атаку вказує серія невдалих входів в систему.	Dictionary attack, SQL injection attack
Атака по збору інформації	Збір інформації або знаходження відомих вразливостей, скануючи або зондуючи комп'ютерні або мережі	SYS scan, FIN scan, XMAS scan
Несанкціоноване підвищення прав користувача до суперкористувача	Такий тип атаки може використовувати вразливості для отримання прав супрекористування в системі при старті в якості звичайного користувача системи. Вразливості включають в себе перехват паролей, атаку по довіднику або соціальну інженерію.	Rookit, loadmodule, perl
Несанкціоноване отримання прав користувача	Здатність відправляти пакети в віддалену систему по мережі, не маючи ніякого облікового запису в цій системі, отримати доступ як у	Warezelient, warezmaster, imap, ftp_wripe, multihop, phf, spyC

Кінець таблиці 1.1

Назва атаки	Характеристика	Приклад
	звичайного користувача або як суперкористувача та виконати шкідливі операції. Також виконання атаки на публічні служби (такі як HTTP та FTP) або під час з'єднання захищених служб (таких як POP та IMAP)	
Зондування	Сканує мережі для визначення працюючих IP-адрес і збирає інформацію про вузол (що це за служба, яка операційна система використовується). Надає зловмиснику список потенційних вразливостей, які згодом можуть бути використані для атаки на окремі системи та служби	IPsweep, portsweep

1.2 Системи виявлення вторгнень (IDS)

Виявлення проникнення в мережу вивчається на протязі майже 20 років. Як правило, поведінка зловмисника помітно відрізняється від поведінки нормального користувача, і, отже, її можна виявити [5]. IDS можна класифікувати залежності від їх розгортання в режимі реального часу: системи на базі хоста (HIDS) та мережеві системи виявлення вторгнень (NIDS).

HIDS – такі системи моніторять та аналізують внутрішню частину обчислювальної системи, а не її зовнішній інтерфейс [6]. Також можуть виявити внутрішню активність, наприклад, яка програма отримує доступ до яких ресурсів та чи пробує ця програма отримати незаконний доступ до системи. Прикладом може слугувати текстовий процесор, який раптово починає модифікувати базу паролей системи. Аналогічним чином, HIDS може перевірити стан системи та її збережену інформацію будь то в оперативній пам'яті чи в файловій системі, в лог файлах або де-небудь ще. Можна думати про такі системи як про агенти, які слідкують за тим, чи не обійшов що-небудь або хто-небудь внутрішню або зовнішню політику безпеки операційної системи.

NIDS – системи призначені для виявлення вторгнень в мережеві дані. Взломи зазвичай відбуваються як аномальні закономірності [7], основною причиною яких є атаки ініційовані зловмисниками, які хочуть отримати доступ до мережі з метою крадіжки інформації або порушення роботи мережі. Зазвичай, мережа підключена до решти світу через Інтернет, в свою чергу, NIDS зчитують всі входящі пакети або потоки, намагаючись знайти підозрілі аномалії. Наприклад, якщо протягом невеликого проміжку часу спостерігається велика кількість з'єднань до дуже великої кількості різних портів, можна припустити, що хтось здійснює атаку «port scan» на якомусь комп'ютері або комп'ютерах в мережі. Різні типи атак «port scan» та програми для їх запуску детально розглянуті у [8]. Окрім перевірки вхідного трафіку, NIDS також надають важливу інформацію про проникнення з вихідного або локального трафіку. Деякі атаки можуть бути навіть ініційовані із середини контрольованої мережі або сегмента мережі, і тому взагалі не розглядаються як вхідний трафік. Данні, доступні для систем виявлення вторгнень, можуть бути на різних транспортних рівнях, наприклад фрейми на пакетному рівні та IPFIX записи. Дані мають велику розмірність, та, як правило, є комбінацією категоріальних та числових атрибутів.

Системи виявлення вторгнень на основі зловживань шукають, зазвичай, відомі моделі вторгнень, а системи на базі виявлення аномалій, навпаки, намагаються виявити незвичайні моделі поведінки. Системи виявлення вторгненьможна розділити на три типи на основі їхнього механізму виявлення: на основі аномалій, на основі сигнатур та гібридні [3], [9], [10]. Детальне порівняння наведене в таблиці 1.2.

Таблиця 1.2 – Типи систем виявлення вторгнень

Тип	Характеристика
На основі сигнатур	Виявлення засновано на наборі правил або сигнатур для кожної із відомих атак. Може виявляти всі відомі моделі атак на основі еталонних даних.
На основі аномалій	Такий тип систем працює на основі головного припущення, що всі вторгнення обов'язково є аномальними. Система створює нормальний профіль діяльності і перевіряє, чи відрізняється стан системи від такого профіля на статистично дуже вагому величину, щоб повідомити про спробу вторгнення. Аномальні дії, які насправді не являються порушенням, можуть бути помічени як незаконні. Це так звані хибнопозитивні результати. Для таких систем потрібно вибирати порогові значення таким чином, щоб не одна із вищенаведених проблем не була необгрунтовано збільшена, а вибір ознак для моніторинга не був оптимізований. Також, таким системам характерна велика обчислювальна складність через великі накладні витрати та, можливо, оновлення декількох матриць системного профілю.
Гібридний	Використовує переваги обох підходів. Намагається виявити як і відомі, так і невідомі атаки

На сьогодні, дослідники в основному зосереджуються на виявлені мережевого вторгнення на основі аномалій, оскільки такий підхід може виявляти невідомі атаки.

Існує декілька причин, чому виявлення вторгнень є необхідною частиною всієї системи захисту. По-перше, багато звичайних систем та додатків розроблювались без уваги до питань безпеки. Такі системи та додатки націлень на роботу в середовищі, де безпека ніколи не була основною проблемою. Однак, ці ж самі системи та додатки, коли вони розгортаються в мережі, стають основним головним болем безпеки. Наприклад, система може бути абсолютно безпечною, коли вона ізольована, але вразлива при підключенні до інтернету. Виявлення вторгнень забезпечує спосіб ідентифікації загрози і, таким чином, дозволяє реагувати на атаки, які здійснюється на системи.

По-друге, в силу обмежених можливостей в області інформаційної безпеки і практики розробки програмного забезпечення, комп’ютерні системи та додатки можуть мати конструктивні недоліки або помилки, які зловмисник може використовувати для атаки на системи або додаток. В результаті, деякі превентивні механізми (наприклад, брандмауери) можуть виявитись не настільки ефективними, як очікувалось.

1.3 Виявлення мережевих аномалій

Виявлення мережевих аномалій являється важливим аспектом в сфері інформаційної безпеки. Виявлення аномалій та зловживань – це альтернативні підходи для виявлення вторгнень в мережу, які входять в системи виявлення вторгнень. Системи виявлення вторгнень складається із трьох основних груп:

виявлення на основі сигнатури, виявлення на основі аномалій та аналіз протоколу. Виявлення аномалій в мережі значно допомагає адміністраторам в керуванні та мережею та усуненню проблем з безпекою.

Системи виявлення аномалій намагаються знайти закономірності в даних, які відрізняються від нормальнох поведінки. Важливість виявлення аномалій обумовлена тим, що аномалії в даних транслюються у важливу (і часто критичну) інформацію, яка може бути використана в найрізноманітніших областях застосування [11]. Наприклад, аномальний трафік в комп'ютерній мережі може означати, що взламаний комп'ютер відправляє конфіденційні дані не авторизованому вузлу мережі. Однак, аномалії в мережі можуть бути викликані декількома різними причинами. Існує дві широкі категорії мережевих аномалій: аномалій, зв'язані із продуктивністю та аномалії, зв'язані із безпекою. Різнимим прикладами аномалій, пов'язаних з продуктивністю, є: широкомовний шторм, раптові перевантаження, збої в роботі мережевих вузлів, підкачування сторінок по мережі та збої в роботі файлового сервера. Мережеві аномалії, зв'язані із безпекою, можуть бути викликані шкідливою діяльністю зловмисників, які навмисно переповнюють мережу непотрібним трафіком для зменшення пропускної здатності мережі, щоб створити створити користувачам проблеми з мережею. Як зазначено у [12], аномалії, пов'язані з безпекою, бувають трьох типів: точкові, контекстні та колективні аномалії, див. табл. 1.3.

Таблиця 1.3 – Типи мережевих аномалій

Тип	Харахтеристика	Приклад
Точкові	Зразок індивідуальних даних, який був визнаний аномальним по відношенню до решти даних	Ізольований екземпляр мережевого трафіку від звичайних нормальних прикладів в певний час.

Кінець таблиці 1.3

Тип	Харахтеристика	Приклад
Контекстні	Екземпляр даних, який був визнаний аномальним в певному контексті. Контекст створюється структурою в наборі даних. Для визначення контексту використовуються два набору атрибутів: контекстуальні і поведінкові атрибути.	Часовий інтервал між покупками при шахрайстві з кредитними картами
Колективні	Набір пов'язаних з прикладів даних, які були визнані аномальними по відношенню до всього набору даних. Набір подій є аномалією, але окремі події не є аномаліями, коли вони відбуваються в послідовності поодинці.	Порядок дій: ...http-web, buffer-overflow, http-web, http-web, ftp, http-web, ssh, http-web, ssh, buffer-overflow ...

На сьогоднішній день виявлення мережевого вторгнення, заснованого на аномаліях, є основним напрямом досліджень і розробок в області виявлення вторгнень. З'являються різні системи з можливостями системи виявлення мережевого вторгнення на основі аномалій (ANDIS), і в даний час вивчаються багато нових підходів. Однак ця тема ще не до кінця опрацьована, і ще належить вирішити ключові питання, перш ніж стане можливим широкомасштабне розгортання ANDIS.

1.4 Загальна архітектура систем виявлення мережевих вторгнень на базі аномалій

Багато архітектур було розроблено дослідниками та спеціалістами-практиками. Однак, розробка ефективної архітектури ANIDS все ще знаходиться на стадії вивчення. Загальна схема архітектури ANIDS показана на рисунку 1.1.

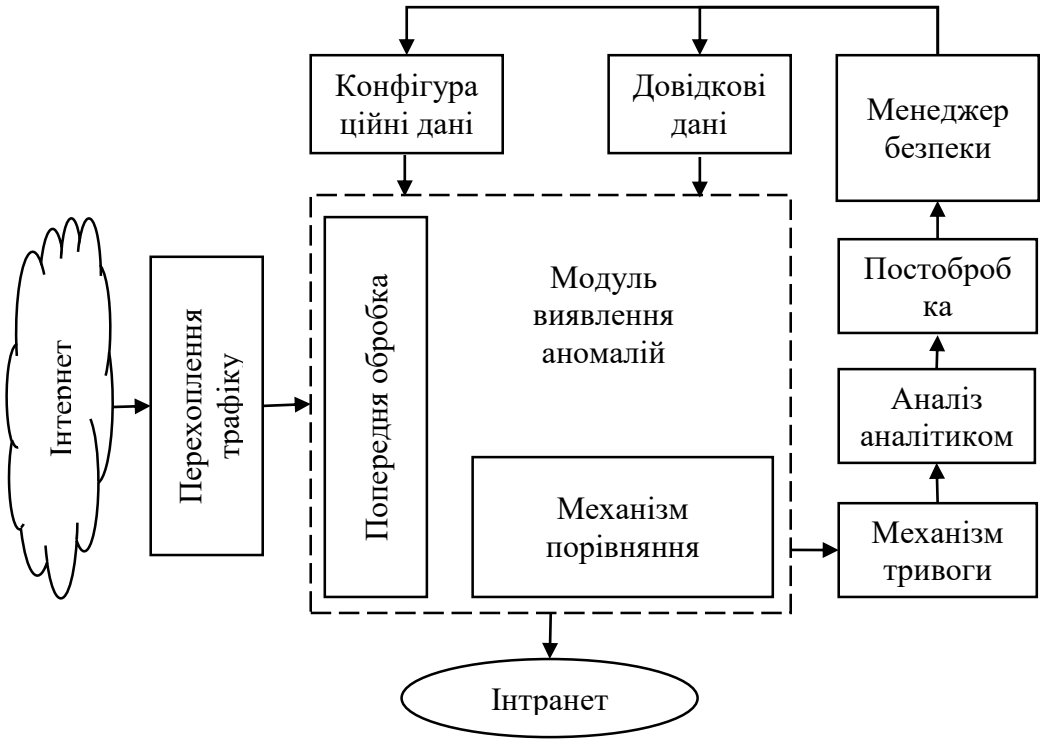

Рисунок 1.1 – Загальна архітектура ANIDS

Архітектура ANIDS складається із таких компонентів:

1. Модуль виявлення аномалій – це основа будь-якої системи виявлення вторгнень. Модуль намагається виявити будь-яке вторгнення як і в режимі онлайн, так і в автономному режимі. Однак, перед тим, як відправити мережевий трафік в модуль виявлення аномалій, його необхідно попередньо обробити. Якщо атаки відомі, то вони модуть бути виявленні за допомогою сигнатурного методу. З іншої сторони, невідомі атаки можуть бути виявлені

за допомогою аномалій, заснованого на механізмі порівняння. Механізм порівняння відповідає за пошук відповідного шаблону або профілю в мережевому трафіку, який може бути побудований за допомогою постійного моніторингу мережевого трафіку, включаючи всі відомі атаки та вразливостію. Для розробки ефективного механізму порівняння, необхідно дотримуватись наступних вимог: порівняння визначає, чи буде новий екземпляр відноситись до відомого класу, заданому профілем високої розмірності чи ні; порівняння повинно бути швидким; ефективна організація профілів може сприяти більш швидкому пошуку при збігу.

2. Довідкові дані – зберігається інформація про сигнатури відомих атак та вразливостей або профілі нормальної поведінки. Довідкові дані повинні зберігатись ефективно. Можливими типами довідкових даних, що використовується в загальній архітектурі NIDS є: профіль, сигнатура та правило. Елементи обробки оновлюються в профілі по мірі того, як з’являються нові знання про поведінку. Такі оновлення проводяться через регулярні проміжки часу та в пакетному режимі.
3. Конфігураційні дані – відповідає за зберігання тимчасових або неповних даних, наприклад, частково створеним сигнатурам вторгнення. Простір, необхідний для зберігання такої інформації, може бути досить великий. Кроки по збільшенню даних конфігурації наведені на рисунку 1.2.

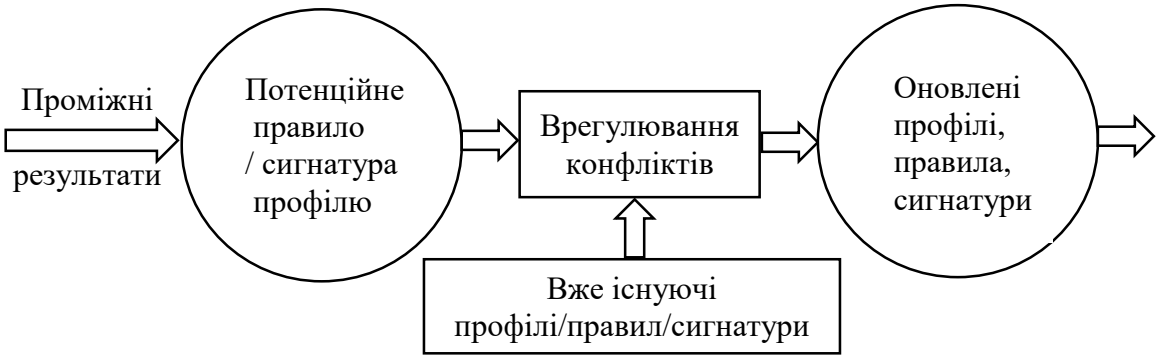

Рисунок 1.2 – Оновлення конфігураційних даних в загальній архітектурі ANIDS

4. Механізм тривоги – відповідає за генерацію сигналів тривоги на основі показників, отриманих від модуля виявлення аномалій.
5. Аналітик – відповідає за аналіз, інтерпретацію та прийняття необхідних заходів на підставі сигнальної інформації, наданої модулем виявлення аномалій. Аналітик також вживає необхідних заходів для діагностики інформації про сигнали тривоги в якості постобробки для підтримки еталонного або профільного оновлення за допомогою менеджера безпеки.
6. Постобробка – модуль для опрацювання згенерованих сигналів тривоги для діагностики реальних атак.
7. Менеджер безпеки – відповідає за оновлення сигнатур вторнень, по мірі того як з’являються нові види атак.

Виявлення мережевих аномалій визначає виявлення аномалій в мережах як проблему пошуку виняткових закономірностей в мережевому трафіку, які не відповідають очікуваній нормальній поведінці. В цьому контексті, закономірності, які виділяються, найчастіше називають аномаліями або відхиленнями.

Методи виявлення мережевих аномалій найчастіше використовують у так званих системах виявлення мережевих аномалій. Системи виявлення мережевих аномалій призначені для обробки мережевих даних шляхом моніторинга пакетів

в мережі та пошуку закономірностей, а також використовуються для класифікації вхідних даних на аномалії та екземпляри нормальних даних. Такі системи працюють в трьох режимах:

а) під наглядом, в якому використовуються тренувальні дані як зі звичайних, так і аномальних класівна;
б) напів керований режим, в якому використовують тільки помічені екземпляри даних для звичайних класів;
в) без нагляду, в якому не потрібні ніякі розмічені приклади даних, та розмітка проводиться самою системою.

Ключовою частиною розробки систем виявлення мережевих аномалій являється розуміння нормальності, яке визначається як формальна модель, яка виражає відносини між фундаментальними змінними, які беруть участь в динаміці системи. Тому, якщо відхилення по відношенню нормального профілю дуже велике, такі випадки класифікуються як аномальні.

Висновок до розділу

В даному розділі описано актуальність задачі, що досліджується у даній магістерській дисертації. Було проаналізовано застосування систем виявлень вторгнень для запобігання атак. Зазначені системи являються основними складовими в забезпеченні безпеки IT інфраструктури. Головна загроза для таких інфраструктур це кібератаки, які постійно розвиваються та яких з кожним днем стає все більше і більше. Що запобігти таким атак, системи виявлення вторгнень аналізують вхідний та вихідний трафік, в результаті чого можна вжити попереджувальні заходи для захисту мережі.

Також було розглянуто дві основні методології, на які можна поділити задачу виявлення вторгнень: на базі аномалій та на базі сигнатур. Після огляду цих підходів, було виявлено, що більш ефективнішим є метод виявлення вторгнень на основі аномалій, так як такий підхід може виявляти як і відомі так і невідомі типи атак, на противагу сигнатурному методу, який ефективно працює тільки проти тих типів атак, які йому вже відомі.

РОЗДІЛ 2 АНАЛІЗ МЕТОДІВ ВИЯВЛЕННЯ МЕРЕЖЕВИХ АНОМАЛІЙ

2.1 Аналітичний огляд задачі виявлення вторгнень

Проблема виявлення вторгнень в мережу – це проблема класифікації або кластеризації, яка складається із наступних компонентів [3]: типи вхідних даних, підходящу міру близькості, розмітка даних, класифікація методів на основі розмічених даних, виявлення вагомих ознак та звіти про аномалії. Типи вхідних даних є ключовим аспектом будь-якого методу виявлення вторгнення в мережу на основі аналізу аномалій, є суттю вхідних даних, що використовуються для аналізу. Вхідними даними зазвичай являються прикладами даних [13]. Кожен приклад даних може бути описаний із використанням набору атрибутів бінарного, категорійного або числового типу. Кожен екземпляр даних може складатись з одного (одновимірного) або декількох(багатовимірного) атрибутів. У випадку багатомірних атрибутів даних, всі атрибути можуть бути одного типу або представляти собою комбінацію типів даних.

Міра близькості (схожість або несхожість) необхідні для вирішення багатьох проблем розпізнавання шаблонів в класифікації та кластеризації. Відстань – це кількісна степінь того, наскільки далеко один від одного два об’єкта. Міри відстані, які задовольняють метричні властивості [13], просто називаються метричними в той час як інші неметрична міра відстані іноді називається розбіжністю. Як правило, міри близькості – це функції, які приймають пару об’єктів як аргументи і повертають числові значення, які стають все більшими по мірі того, як об'єкти стають більш схожими. Міра близькості S зазвичай визначається як функція:

$$X \times X \to \mathbb{R} \tag{2.1}$$

що відповідає наступними властивостями [14]:

а) функція додатна:

$$\forall_{x,y} \in X, S(x,y) \geq 0 \tag{2.2}$$

б) функція симетрична:

$$\forall_{x,y} \in X, S(x,y) = S(y,x) \tag{2.3}$$

в) функція максимальна:

$$\forall_{x,y} \in X, S(x,x) \geq S(x,y) \tag{2.4}$$

де X це простір даних, а x, y – пара k-мірних об'єктів.

Найбільш поширені формули відстаней для числових [15]-[27], категорійних [28] та даних змішаного типу [29] наведено відповідно в таблицях 2.1, 2.2, 2.3

Таблиця 2.1 – Формули відстаней для числового типу даних

Назва	Відстань, $S_i(x_i, y_i)$
Евклідова відстань	$\sqrt{\sum_{i=1}^{d} \lvert x_i - y_i \rvert^2}$
Квадрат евклідової відстані	$\sum_{i=1}^{d} \lvert x_i - y_i \rvert^2$

Кінець таблиці 2.1

Відстань Мінковського	$\sqrt[p]{\sum_{i=1}^{d} \lvert x_i - y_i \rvert^p}$
Відстань Жаккара	$\frac{\sum_{i=1}^{d} x_i\, y_i}{\sum_{i=1}^{d} x_i^2 + \sum_{i=1}^{d} y_i^2 - \sum_{i=1}^{d} x_i\, y_i}$
Відстань Чебишева	$\max_i \lvert x_i - y_i \rvert$

Таблиця 2.2 – Формули відстаней для категорійного типу даних

$w_k, k = 1 \dots d$	Відстань, $S_k(x_k, y_k)$
$\frac{1}{2}$	Перетин = $\begin{cases} 1 \text{ якщо } x_k = y_k \\ 0 \text{ інакше} \end{cases}$
$\frac{1}{d}$	$IOF = \begin{cases} 1 & \text{якщо } x_k = y_k \\ \frac{1}{1 + \log f_k(x_k) x \log fk\,(y_k)} & \text{інакше} \end{cases}$
$\frac{1}{d}$	$Eskin = \begin{cases} 1 & \text{якщо } x_k = y_k \\ \frac{n_k^2}{n_k^2 + 2} & \text{інакше} \end{cases}$
$\frac{1}{d}$	$OF = \begin{cases} 1 & \text{якщо } x_k = y_k \\ \frac{1}{1 + \log \frac{N}{f_k(x_k)} x \log \frac{N}{f_k(y_k)}} & \text{інакше} \end{cases}$

Таблиця 2.3 – Формули відстаней для змішаного типу даних

Назва	Формула відстані
Загальний коефіцієнт подібності	$s_{gsc}(x, y) = \frac{\sum_{k=1}^{d} w(x_k, y_k)\, s(x_k, y_k)}{\sum_{k=1}^{d} w(x_k, y_k)}$ Для числових атрибутів, $s(x_k, y_k) = 1 - \frac{\lvert x_k - y_k \rvert}{R_k}$,

Кінець таблиці 2.3

Назва	Формула відстані
	де R_k знаходиться в діапазоні k атрибуту; $w(x_k, y_k) = 0$ якщо x або y мають пропущені значення для атрибуту k, інакше $w(x_k, y_k) = 1$ Для категорійних атрибутів, $s(x_k, y_k) = 1$ якщо $x_k = y_k$, інакше $s(x_k, y_k) = 0$; $w(x_k, y_k) = 0$ якщо точки даних x або y мають пропущені значення для атрибуту k, інакше $w(x_k, y_k) = 1$
Загальний коефіцієнт відстані	$d_{gsc}(x, y) = \left(\frac{\sum_{k=1}^{d} w(s_k, y_k)\, d^2(x_k, y_k)}{\sum_{k=1}^{d} w(x_k, y_k)} \right)^{\frac{1}{2}}$
	де $d^2(x_k, y_k)$ це відстань атрибута k в квадраті; для числових атрибутів, $d(x_k, y_k) = \frac{\lvert x_k - y_k \rvert}{R_k}$, де R_k знаходиться в діапазоні k атрибуту; $w(x_k, y_k) = 0$ якщо x або y мають пропущені значення для атрибуту k, інакше $w(x_k, y_k) = 1$ для категорійних атрибутів, $d(x_k, y_k) = 0$ якщо $x_k = y_k$, інакше $s(x_k, y_k) = 1$; $w(x_k, y_k) = 0$ якщо точки даних x або y мають пропущені значення для атрибуту k, інакше $w(x_k, y_k) = 1$

Для числових значень передбачається, що дані представлені у вигляді дійсних векторів. Атрибути беруть свої значення із неперервного діапазону. В наведених вище формулах відстані для числового типу передбачається, що є два

об’єкта $x = x_1, x_2, x_3, \ldots x_d$, $y = y_1, y_2, y_3, \ldots y_d$ та Σ^{-1} представляє собою коваріацію даних із d атрибутів, тобто розмірність.

Для категорійних даних обчислення схожості або відстані вимірів не є однозначним в силу того, що немає чіткого поняття порядку між категорійними значенням. Найпростіший спосіб знайти схожість між двома категорійними атрибутами – це присвоїти 1, якщо значення ідентичні, або 0 якщо значення не ідентичні. В таблиці 2.2, відстань $S_k(x_k, y_k)$ означає схожість між атрибутами. Вага атрибуту w_k для атрибуту k обраховується так, як показано в таблиці. Для категорійного датасету D, який містить n – об’єктів, визначених через набір d Категорійних атрибутів, де A_k обозначає k атрибут. В такому випадку, $S_k(x_k, y_k)$ визначає для кожного атрибуту близькість між двома значеннями для категорійного атрибуту A_k, $x_k, y_k \in A_k$.

Змішаний тип даних може містити в собі значення як і числового так і категорійного типу даних. Поширеною практикою при кластеризації та класифікації змішаного набору даних є перетворення категорійних значень в числові а потім вже використання числових алгоритмів. Інший підхід заключається в прямому порівнянні категорійних значень, при якому два різних значення означають відстань 1, в той час як ідентичні значення – відстань 0. Дві широко відомі міри близькості, загальний коефіцієнт подібності та загальний коефіцієнт відстані [29] для змішаного типу даних наведені в таблиці 2.3. Такі методи можуть не враховувати інформацію про подібність, закладену в категорійних значеннях. Відтак, цільовий алгоритм може не зовсім точно виявити структуру подібності в наборі даних [29], [30].

Розмітка даних. Мітка, зв’язана з прикладом даних, визначає, чи являється цей конкретний приклад даних нормальним або аномальним. Слід зазначити, що отримання точних розмічених даних як нормальних так і аномальних прикладів часто є вкрай затратним. Розмітка часто виконується спеціалістами вручну, і тому

потрібно чималі зусилля для отримання точно розміщеного начального набору даних [3]. Більш того, аномальна поведінка часто носить динамічний характер, наприклад, можуть виникати нові типи аномалій, для який не існує розмічених тренувальних даних.

Класифікація методів на основі розмічених даних. В залежності від доступності розмічених даних, методи класифікації даних можуть працювати в трьох режимах: навчання з учителем, напівавтоматичне навчання, навчання без учителя.

В режимі навчання з учителем мається на увазі наявність навчального набору даних, який позначає екземпляри як для нормального так і для аномального класу. Типовий підхід в таких випадках полягає в побудові прогностичної моделі для нормальних та аномальних класів. Будь який невідомий екземпляр даних порівнюється з моделлю, щоб визначити, до якого класу він належить. При виявленны аномалій за допомогою методів навчених із учителем, виникають дві основні проблеми. По-перше, аномальних випадків набагато менше в порівнянні із звичайними прикладами в навчальних даних. Проблеми, які виникають через незбалансоване розподілення класів, були розглянуті в літературі по інтелектуальному та машинному навчання [31]. По-друге, отримання точних даних та репрезентативних міток, особливо для класу аномалій, зазвичай є складною задачею. Ряд методик вводять штучні аномалії в звичайний набір даних для отримання розміченого набору навчальних даних[32].

Методи напівавтоматичного навчання припускають, що навчальні дані були помічені екземплярами тільки для нормального класу. Так як вони не вимагають позначок для класу аномалій, вони можуть бути використані більш ефективно, порівняно з методиками, які перебувають під наглядом. Наприклад, при виявленні несправностей космічних апаратів [33], сценарій аномалії означатиме аварію, яку нелегко змоделювати. Типовий підхід, який

використовується в таких методиках, полягає в тому, щоб побудувати модель для класу, відповідного нормальної поведінки, і використовувати модель для ідентифікації аномалій в даних випробуваннях.

Методи навчання без учителя не вимагають початкових даних для навчання, та потенційно являються найбільш широко застосовуваними. Методики, що відносять до цієї категорії, припускають, що звичайні випадки зустрічаються набагато частіше чим аномалії в даних тестування [34]. У тих випадках, коли це припущення не відповідає дійсності, такі методи страждають від високої частоти помилкових спрацьовувань. Більшість методів напівавтоматичного навчання можуть бути адаптовані для роботи в режимі навчання без учителя із використанням в якості тренувальних даних вибірки нерозмічених даних [35]. Така адаптація передбачає, що дані тестів містять дуже мало аномалій, і модель, отримана в ході навчання, стійка до цих нечисленних аномалій.

Виявлення вагомих ознак. Відбір ознак відіграє важливу роль в виявленні мережевих аномалій. Методи відбору ознак використовуються в області виявлення вторгнень для усунення неважливих або неактуальних ознак. Відбір ознак зменшує обчислювальну складність, усуває надмірність інформації, підвищує точність алгоритму виявлення, полегшує розуміння даних та покращує універсальність даних. Процес відбору характеристик складається з трьох основних етапів: формування підмножини, оцінка підмножини та перевірка достовірності. Існує три різних підходи до генерації підмножини ознак: повний, евристичний та випадковий. Функції оцінки розділені на п’ять [36] окремих категорій: на основі балів, на основі ентропії або загальної інформації, на основі кореляції, послідовності та точності виявлення. Моделювання та практична реалізація – це два шляхи для перевірки підмножини, яка оцінюється. Концептуальна основа процесу відбору підмножини показана на рисунку 2.1.

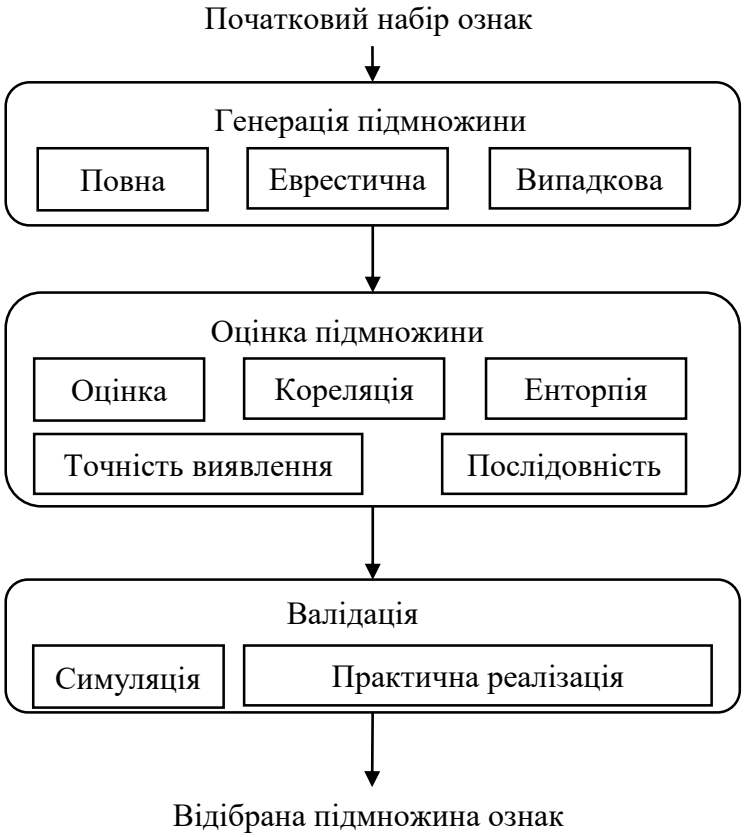

Рисунок 2.1 – Структура процесу відбору підмножини ознак

Методи відбору ознак поділяються на три типи: фільтровий, обгортковий та гібридний [7]. На відміну від обгорткових методів, які намагаються оптимізувати деякі зумовлені критерії щодо набору ознак в рамках процесу вибору [37], методи фільтрового типу покладаються на загальні характеристики навчальних даних для вибору ознак, які не залежать одне від одного але сильно залежать від результату [38]. Гібридний метод вибору характеристик намагається використовувати основні особливості як і обгорткових так і фільтрових методів. Декілька інших методів для обирання ознак наведено в [39], [40] – [42].

Звіти про аномалії. Важливим аспектом будь-якого методу виявлення аномалій є спосіб звітування про наявність аномалії [3]. Як правило, результати, які отримуються за допомогою методів виявлення аномалій, бувають двох типів: оцінка, яка представляє собою значення, що поєднує відстань та відхилення із посиланням на набір профілів або підписів, вплив більшості в своєму оточенні

та явне домінування відповідного підпростору; мітка, яка є значенням (аномалія чи нормальна поведінка), та яка надається кожному прикладу окремо. Зазвичай, розмітка прикладу залежить від розміру груп, які створені методом навчання без учителя, щільності групи(груп) , явного домінування подмножини ознак.

2.2 Методи та системи виявлення мережевих аномалій

Класифікація методів та систем виявлення мережевих аномалій показано на рисунку 2.2.

Рисунок 2.2 – Класифікація методів виявлення мережевих аномалій

В цьому дослідженні проводиться відмінність між виявлення мережевих аномалій та системами, хоча іноді їх поділити між собою дуже складно. В наведеній схемі класифікації (рис. 2.2), методи та системи класифікувались по природі використовуваних алгоритмів. Однак, придумати класифікаційну схему

для методів та систем не так просто, в першу чергу, через те, що методи, які використовуються в різних класах, дуже часто співпадають між собою. Системи виявлення мережевого вторгнення зазвичай інтегрують метод виявлення мережевого вторгнення в архітектуру, яка включає в собі інші пов'язані підсистеми для створення автономної практичної системи, яка може виконувати весь спектр дій, необхідних для виявлення вторгнення.

Статистичні методи та системи. В статистиці, аномалія – це спостереження, яке, як передбачається, частково або повністю несуттєве, оскільки воно не генеруються стохастичною моделлю. Зазвичай, статистичні методи підганяють статистичну модель (для нормальної поведінки) під дані, а потім застосовують статистичний аналіз для визначення того, чи належить до цієї моделі невідомий приклад даних. Приклади, які з малою ймовірністю можуть бути отримані з отриманої моделі, являються аномаліями. Для розробки статистичних моделей виявлення аномалій можна застосувати як параметричні, так і не параметричні методи. У той час як параметричні методи припускають наявність знань про вихідний розподіл та оцінюють параметри на основі отриманих даних [43], непараметричні методи, як правило, не передбачають наявність знань про вихідний розподіл.

Як приклад статистичної IDS, можна розглянути HIDE [44]. HIDE – це основана на аномаліях мережева система виявлення вторгнень, яка використовує статистичні моделі та нейромережеві класифікатори для виявлення вторгнень. HIDE – це розподілена система, яка складається із декількох рівнів, кожен із яких містить декілька агентів виявлення вторгнень (IDA). IDA – це компоненти IDS, які відслідковують діяльність хоста або мережі. Загальну архітектуру системи HIDE наведено на рисунку 2.3.

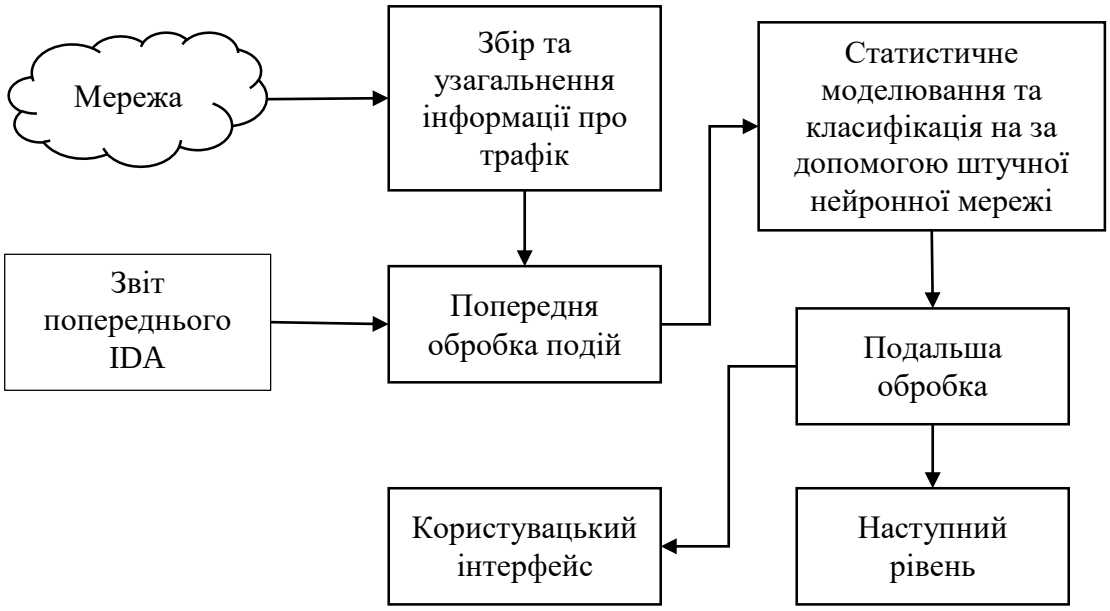

Рисунок 2.3 – Архітектура HIDE системи

Рівень дослідження (верхній рівень на рисунку 2.3) збирає мережевий трафік на вузлі або в мережі, узагальнює трафік в набір статистичних змінних для відображення стану мережі та періодично генерує звіти для попередньої обробки подій. Рівень попередньої обробки подій отримує звіти як від рівня дослідження так і від IDA на нижніх рівня, та перетворює інформацію в формат вхідних даних для статистичної моделі. Статистична препроцесор містить еталонну модель типової мережевої активності, порівнює звіти із рівня попередньої обробки подій та еталонними моделями та формує керуючий вектор для подачі в нейромережевий класифікатор. Класифікатор на основі штучної нейронної мережі аналізує керуючий вектор від статистичної моделі для вирішення питання про те, наскільки поточний мережевий трафік є нормальним. Рівень подальшої обробки генерує звіти для агентів більш високих рівнів. Головною особливістю системи HIDE є здатність виявляти UDP-сміття навіть при інтенсивності атаки до 10% від фонового трафіку.

Баєсова мережа здатна виявити аномалії в мультикласовій системі. Основна методика передбачає незалежність між різними атрибутами. Також,

дослідниками було запропоновано декілька варіантів базової методології [3], які захоплюють умовні взаємозв'язки між різними атрибутами, використовуючи більш складніші Баєсових мереж. На основі такої технології, було запропоновано схему виявлення вторгнення на основі класифікації із використанням Байсових мереж. Байєсівський підхід прийняття рішень покращує якість виявлення, що значно знижує кількість хибно позитивних результатів.

Для виявлення однокласової аномалії також можна використати простий метод інтелектуального пошуку правил асоціації, який базується на підрахунку кількості повторень елементів в базах даних транзакцій, шляхом генерації правил із даних в режимі навчання без учителя. Найбільш складною частиною алгоритму виявлення асоціацій являється пошук наборів елементів, які мають найсильнішу підтримку. Також було представлено алгоритм LERAD, який навчає правила пошуку рідкісних подій в даних часових рядах, з великою залежністю від дальності та знаходить аномалії в мережевих пакетах по TCP-сесіям. LERAD використовує алгоритм, схожий на алгоритм Apriori, який знаходить умовні правила по номінальним атрибутам в числовому ряді, наприклад, послідовність вхідних клієнтських пакетів. Попередником створеного правила являється, а останнім – набір допустимих значень, наприклад port=80 та word3=HTTP/1.0, то word1=GET або POST. Значення дозволено, якщо воно спостерігається хоча б в одному навчальному екземплярі, яке задовольняє попередні умови. Ідея полягає в тому, щоб виявити рідкісні аномальні події; ті, які не відбувалися довгий час і які мають високий бал аномалії. LERAD – це алгоритм із двох частин. В першій частині із випадкової вибірки навчальних даних мережевого трафіку в якому відсутні приклади атак, генерується набір правил-кандидатів. В другій частині алгоритму правила навчаються шляхом отримання набору дозволених значень для кожного попередника.

Алгоритм виявлення аномалій на основі корисного навантаження PAYL намагається виявити найпершу появу хробака або в шлюзі мережевої системи, або у внутрішній мережі від неавторизованого пристрою та запобігти його розповсюдженню. Такий алгоритм використовує не залежну від мови n-грамову статистичну модель вибірки потоків даних.

В доповненні до методів виявлення, існує декілька статистичних NIDS. Як зазначалось раніше, NIDS включають в себе один або декілька методів виявлення вторгнень, які інтегруються с іншими потрібними підсистемами, необхідними для створення підходящої системи. Як приклад такого підходу, можна розглянути систему FSAS (Flow-based Statistical Aggregation Scheme), яка представляє собою потокову статистичну IDS. Вона складається із двох модулів: генератора ознак та детектора на основі потоку. В генераторі ознак, модуль попередньої обробки подій збирає мережевий трафік хоста або мережі. Після обробки подій, відповідні агенти формують звіт для модуля управління потоком. Модуль управління потоками ефективно визначає, чи являється пакет частиною існуючого потоку чи він повинен генерувати новий ключ потоку. Перевіряючи потокові ключі, модуль агрегує потоки разом та динамічно оновлює вимір для кожного потоку. Модуль часу подій періодично викликає модуль відбору ознак для перетворення потоків в необхідний статистичній моделі формат. Класифікатор на основі нейронних мереж класифікує вектори оцінки для визначення пріоритету потоків із рівнем злоякісності. Чим вище злоякісність потоку, тим вище ймовірність того, що цей потік атакуючий та спрямований на зараження або виведення з ладу інфраструктури.

Окрім їх властивостей до виявлення мережевих аномалій, статистичні підходи мають ряд додаткових серйозних переваг:

- Вони не потребують попередніх знань нормальної діяльності цільової системи. Замість цього, вони мають можливість дізнатись очікувану поведінку системи із спостережень.
- Статистичні методи можуть забезпечити точне сповіщення або генерування сигналів тривоги про шкідливі дії, які відбуваються на протязі довгих періодів часу, при умові встановлення відповідного порогу або налаштувань параметрів.
- Вони аналізують трафік, грунтуючись на теорію різких змін, тобто стежать за трафіком на протязі всього часу і повідомляють про будь-які різкі зміни (тобто значних відхиленнях).

Недоліки статистичних моделей виявлення мережевих аномалій наступні:

- Вони сприятливі до навчання атакуючого таким чином, що мережевий трафік який генерується під час атаки, вважається нормальним.
- Встановлення значень різних параметрів або метрик являється складною задачею, особливо через те, що проблема заключається в балансі між хибно позитивними та хибно негативними результатами. Більш того, передбачається статистичний розподіл змінної, але не всі моделі поведінки можна змоделювати стохастичними методами. Більш того, більшість схем спираються на припущення про квазістаціонарний процес, що не завжди реалістично [45].
- Для того, щоб вперше повідомити про аномалії, потрібно багато часу, так як побудова моделі займає тривалий час.
- Для виявлення аномалій можуть бути застосовані статистичні дані по декільком гіпотезам. Вибір найкращої статистики часто не являється простим. Зокрема, побудова гіпотезних тестів для складних розподілень, які необхідні для підбору високорозмірних наборів даних, являється не тривіальною задачею.

- Методи, засновані на гістограмах, відносно прості в реалізації, але ключовим недоліком таких методологій для багатовимірних даних являється те, що вони не здатні зафіксувати взаємодію між атрибутами.

Методи та системи, що базуються на класифікації. Класифікація – це проблема визначення того, до якої категорії відноситься новий приклад даних, на основі навчального набору даних, який містить розмічені приклади, тобто до якої категорії відноситься кожен приклад. Нехай, два класса, приклади яких показані як + та -, та кожен об'єкт може бути визначений у вигляді двух атрибутів або ознак x_1 та x_2, лінійна класифікація намагається знайти лінію між класами як показано на рисунку 2.4.

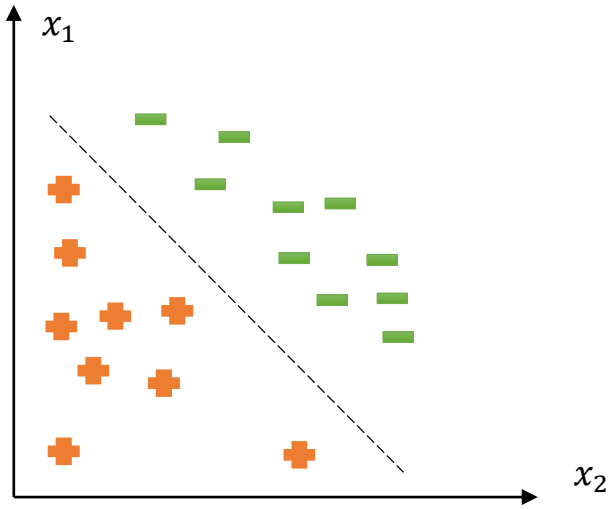

Рисунок 2.4 – Лінійна класифікація

Межі класифікації не завжди можуть бути лінійними, як показано на рисунку 2.5

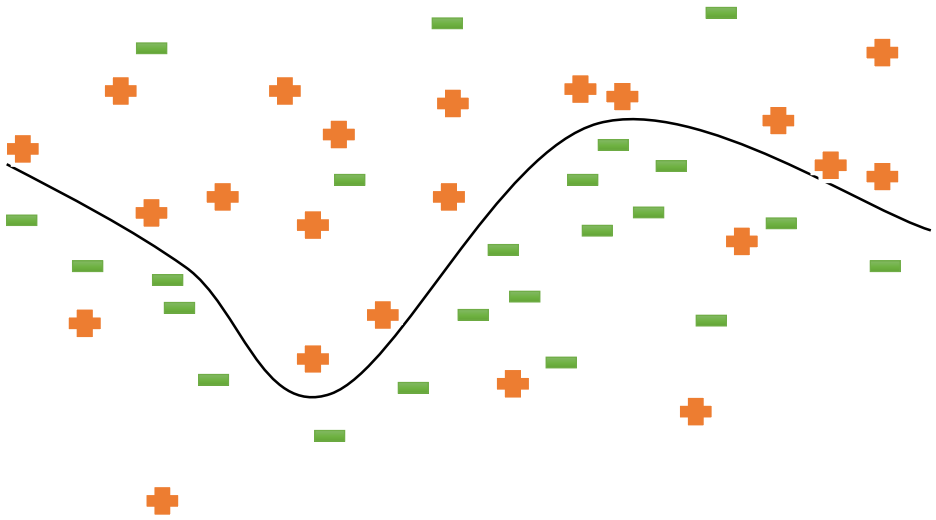

Рисунок 2.5 – Нелінійна класифікація

При виявленні виявленні вторгнень, дані мають високу розмірність, а не тільки два атрибути. Ознаки, як правило, являються змішаними, числовими та категорійними, як було зазначено вище. Таким чином, методи класифікації засновані на створені явної або неявної моделі моделі, що дозволяє категоризувати шаблони мережевого трафіку на декілька класів. Відмінною особливістю цих методик є те, що вони потребують розмічених даних для навчання поведінкової моделі, що пред'являє високі вимоги до ресурсів. В багатьох випадках застосування принципів машинного навчання, таких як класифікація, збігається зі статистичними методами, хоча перша методитка орієнтована на побудову моделі, що підвищує її ефективність на основі попередніх результатів [46]. Для виявлення аномалій в мережевому трафіку було застосовано кілька класифікаційних методів (наприклад метод k-найближчих сусідів, метод опорних векторів та дерева рішень). Прикладом IDS на основі класифікацій є система Automated Data Analysis and Mining(ADAM) [47], яка представляє тестовий стенд для виявлення аномальних випадків. Архітектурна схема ADAM показана на рисунку 2.6.

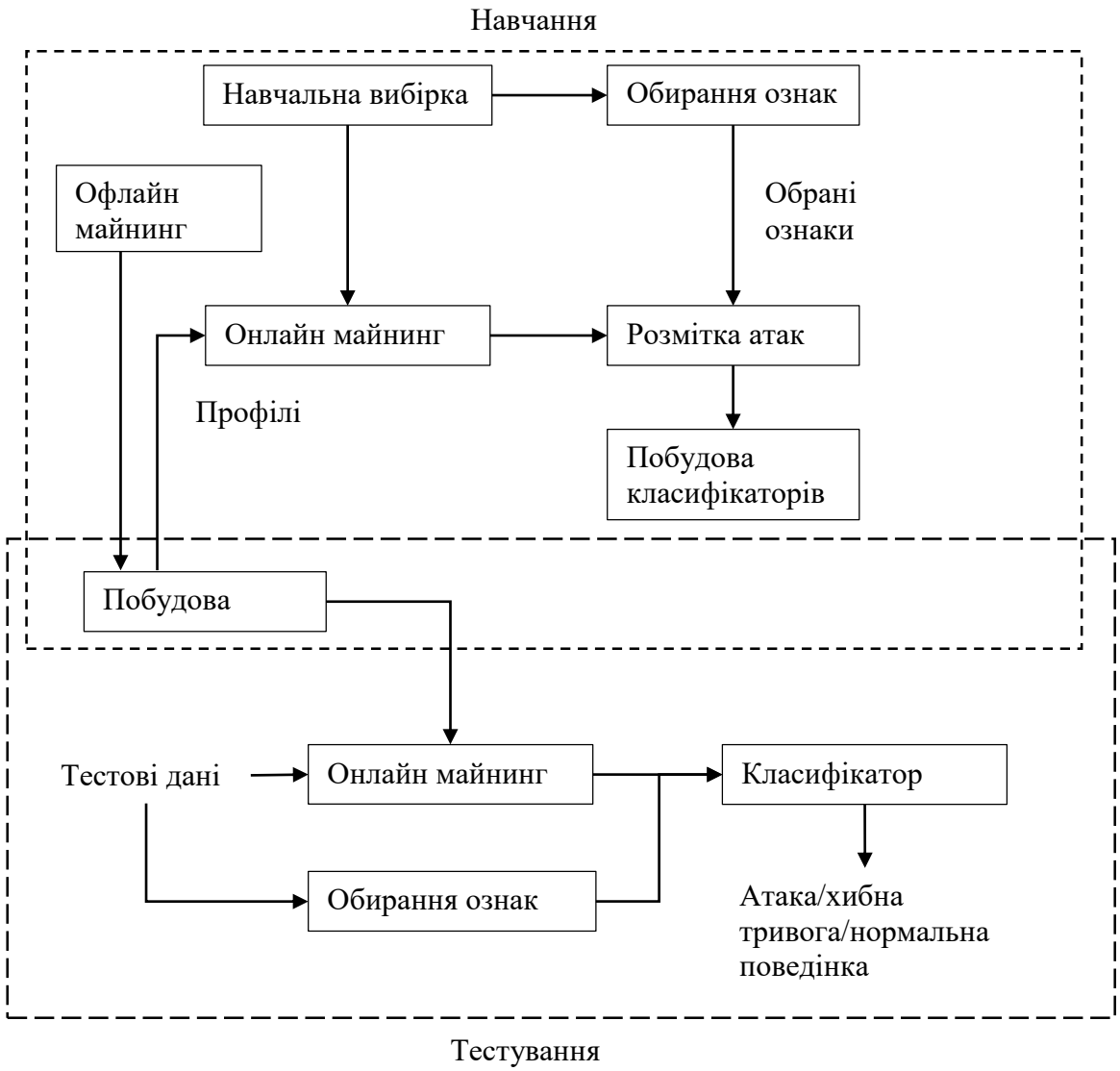

Рисунок 2.6 – Архітектура ADAM системи

ADAM використовує комбінацію технік класифікації та пошуку правил асоціації для виявлення атак в аудиторському журналі мережевого трафіку. Спершу, ADAM будує сховище звичайних наборів частих едемнтів із часових проміжків в яких не було виявлено атак. Далі, ADAM запускає онлайн-алгоритм на основі ковзаного вікна, яке знаходить поширені набори елементів в з'єднання та порівнює їх с тими, які зберігаються в сховищі звичайних наборів елементів, відкидаючи ті, які вважаються нормальними. ADAM використовує класифікатор, який був навчений класифікувати підозрілі з’єднання як відомий тип атаки, невідомий тип атаки або як хибну тривогу.

Методи виявлення аномалій, засновані на класифікації, зазвичай дають кращі результати, ніж методи навчання без учителя (наприклад, які базуються на кластеризації), через те, що використовуються розмічені навчальні дані. В традиційній класифікації нова інформація може бути включена шляхом перенавчання всього набору даних. Однак, це займає багато часу. Інкрементні алгоритми класифікації роблять таке навчання більш ефективним. Хоча методи, засновані на класифікації, популярні, вони не можуть виявити або передбачити невідому атаку або подію до тих пір, поки відповідна інформація про навчання не буде подана для перепідготовки.

Метод опорних векторів являються дуже успішним максимально розділовим класифікатором. Однак, метод опорних векторів займає багато часу для навчання, коли набір даних дуже великий. Для зменшення тривалості навчання при класифікації великих даних вторгнення, було запропоновано використати ієрархічний метод кластеризації, який називається Динамічно зростаюче саморганізаційне дерево (Dynamically Growing Self-Organizing Tree – DGSOT), яке чергується із методом опорних векторів. DGSOT, оснований на штучний нейронних мережах, векористовується для пошуку граничних точок між двома класами. Граничні точки являються найбільш підходящими точками для навчання методу опорних векторів. В методі опорних векторів обчислюються максимальні межі, які розділяють два класи точок даних. На обчислення цих меж впливають тільки точки, які називаються опорними векторами. Решта точок можуть бути відкинуті, не впливаючи на кінцевий результат.

Підходи для виявлення мережевих аномалій засновані на класифікації є доволі популярними. Нижче перераховані деякі переваги таких підходів:

- Ці методи являються дуже гнучкими для навчання та тестування. Вони здатні оновлювати свої стратегії виконання із введенням нової інформації. Відтак, для таких методів адаптація є можливою.

- В них висока швидкість виявлення відомих атак при відповідній установці порогового фзначення.

Хоча такі методи популярні, в них є наступні недоліки:

- Методи в значній мірі залежать від припущень, зроблених класифікаторами.
- Вони споживають більше ресурсів, ніж інші методи.
- Вони не можуть виявити або передбачити невідомий напад або подію до тих пір, поки не буде подана відповідна навчальна інформація.

Кластеризація – це задача розподілення набору об’єктів по групам, які називаються кластерами, щоб об’єкти в одному і в тому ж кластері були більш-менш схожі один на одного, чим на об’єкти в інших кластерах. Кластеризація використовується в розвідувальному добуванню даних. Наприклад, якщо існує набір нерозмічених об’єктів в двох вимірах, то їх можна згрупувати в 5 кластерів, намалювавши навколо їх круги або еліпси, як показано на рисунку 2.7, де C_i – кластери.

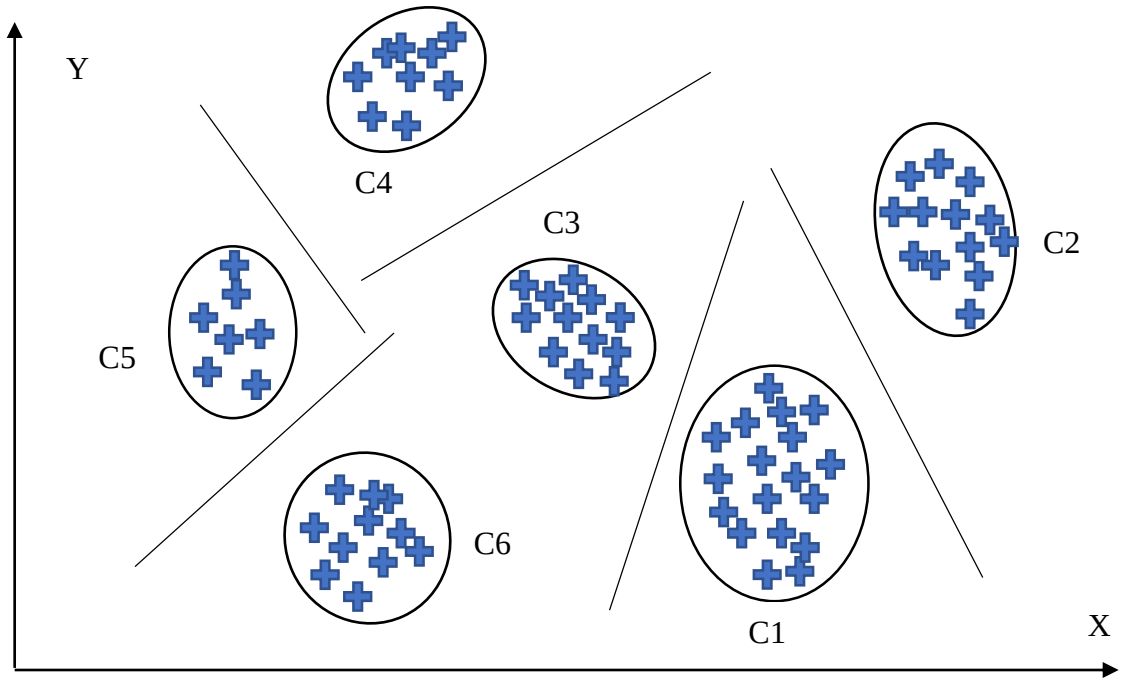

Рисунок 2.7 – Кластеризація в двовимірному просторі

Викиди - це ті точки в наборі даних, які малоймовірні при моделюванні даних, як показано на рисунку 2.8, де C_i – кластери, O_i – викиди.

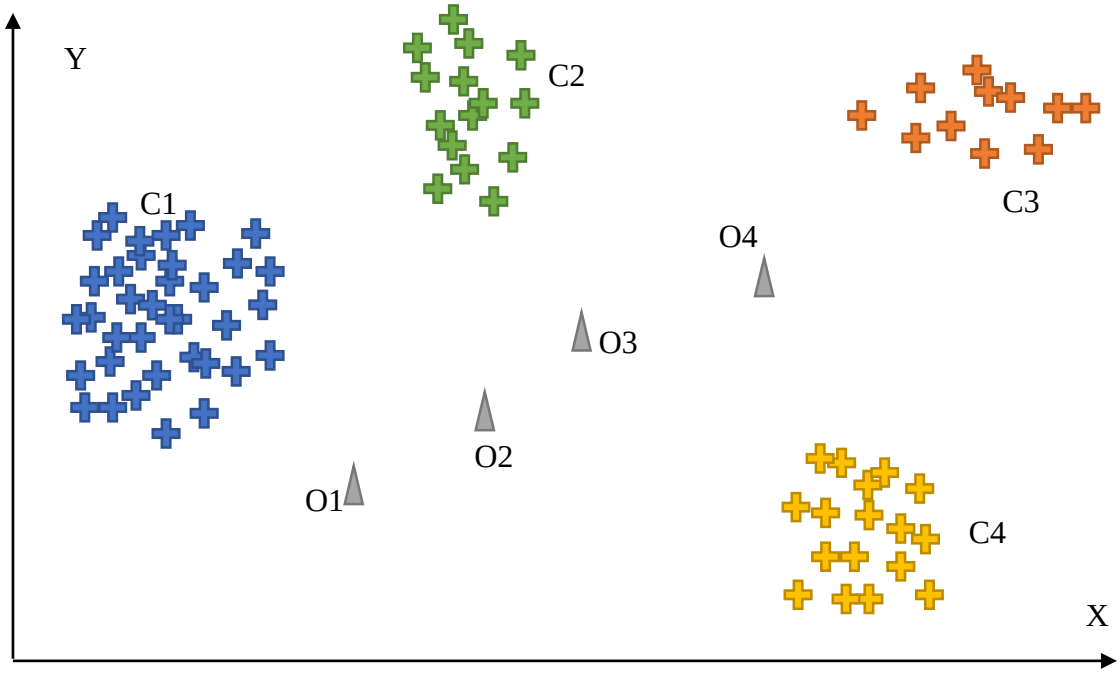

Рисунок 2.8 – Викиди в двовимірному просторі

Кластеризація та пошук викидів є прикладами методів машинного навчання без учителя.

Кластеризація може використовуватись для виявлення мережевих аномалій в автономному середовищі. Такий підхід приносить додатковий ступінь захисту для адміністраторів та дозволяє їм більш точно визначати загрози для їх мережі за допомогою використання набору методів на даних з декількох джерел. Таким чином, великий обсяг діяльності, який може знадобитися для виявлення вторгнення в режимі реального часу в онлайн NIDS, можна прибрати без зменшення ефективності системи.

Наприклад, MINDS (Minnesota Intrusion Detection System) [48] – система виявлення вторгнень на основі добування даних. Архітектура системи показана на рисунку 2.9.

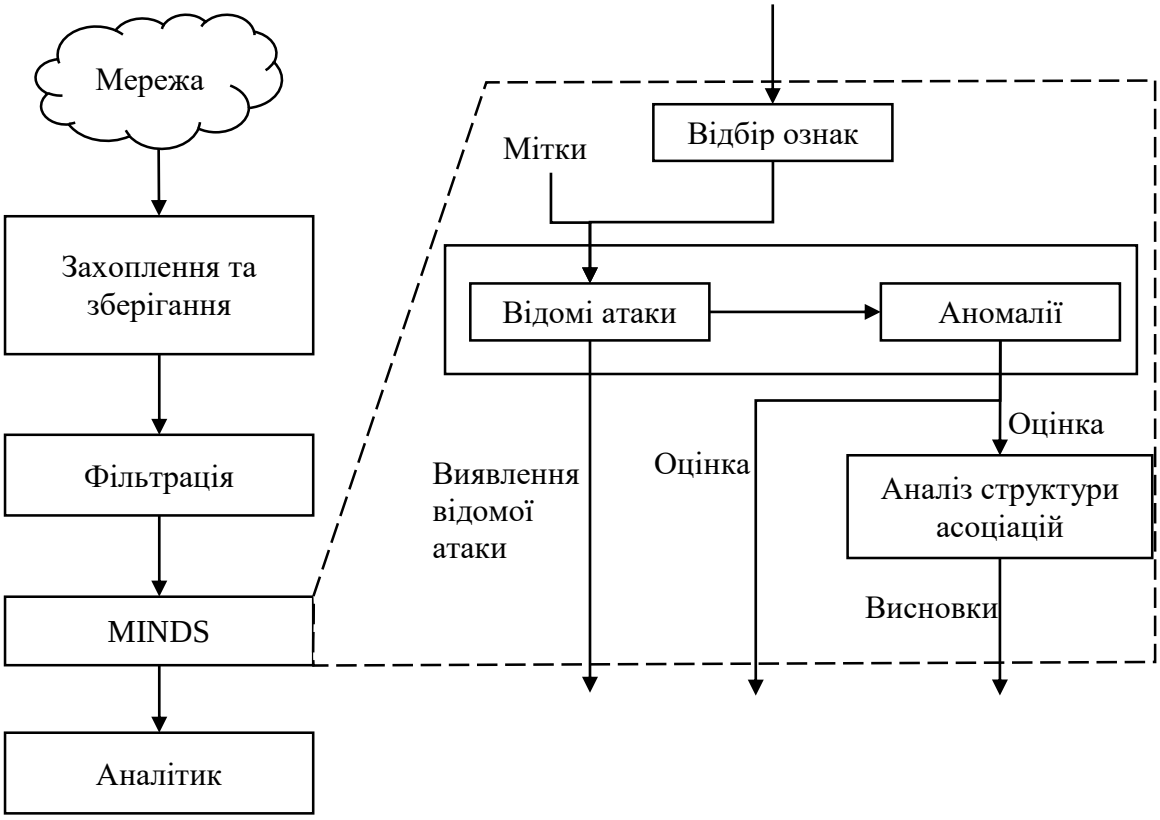

Рисунок 2.9 – Архітектура системи MINDS

В якості вхідних даних, система приймає дані NetFlow, зібрані за допомогою потокових інструментів. Потокові інструменти захоплюють тільки інформацію про заголовкі пакетів і будують однонаправлені сесії потоків. Далі, аналітик використовує результати роботи системи MINDS для аналізу цих файлів даних в пакетному режимі. Причина запуску системи в пакетному режимі полягає не в тому, що на аналіз цих файлів витрачається час, а в тому, що аналітику так буде зручніше. Перед подачею даних в модуль виявлення аномалій, виконується етап фільтрації даних для видалення трафіку, в якому аналітик не зацікавлений.

Першим кроком MINDS є відбір необхідних ознак, які використовуються. Далі, сумуються ознаки спираючись на часові інтервали. Після етапу побудови ознак, модуль виявлення відомих атак використовується для виявлення мережевих з’єднань, які можуть відповідати атакам та для яких доступні

сигнатури, та видалення їх для подальшого аналізу. Далі використовується методика пошуку викидів для присвоєння рейтингу кожному мережевому з'єднанню. Аналітик отримує та переглядає на найаномальніші з’єднання, щоб визначити, чи є вони реальними атаками чи представляють собою іншу цікаву поведінку. Модуль аналізу структури асоціації цієї системи призначений для узагальнення мережевих підключень у відповідності до присвоєного рейтингу аномалії. Аналітик надає зворотній зв’язок після аналізу створених узагальнень та приймає рішення про те, чи корисні ці узагальнення для створення нових правил, які можуть бути використані в виявленні атак.

Методи кластеризації часто використовують при виявленні аномалій. До них відносяться алгоритми однозвязної кластеризації – метод найближчих сусідів та ієрархічні алгоритми кластеризації.

Деякі переваги використання кластеризації:

- Якщо наперед відоме число кластерів, на які потрібно розбити дані, то задача кластеризації стає не дуже складною.
- Інкрементальна кластеризація (в режимі навчання із учителем) ефективна для генерації швидкої відповіді.
- Для виявлення мережевих аномалій вигідно у випадку із великим набором даних згрупувати їх в однакову кількість класів або кластерів, так як це знижує обчислювальну складність при виявленні вторгнень.
- Забезпечує стабільну роботу порівняно із класифікаторами або статистичними методи.

Недоліки методів, основаних на кластеризації:

- Більшість методик були запропоновані для рорити з безперервними атрибутами.

- В методиках виявлення вторгнень на основі кластеризації передбачається, що більші кластери є нормальними, а дрібніші – атакою чи вторгненням. Без цього припущення важко оцінити методику.
- Використання непідходящої міри наближення негативно впливає на швіидкість виявлення.
- Динамічне покращення профілів займає багато часу.

Методи та системи на основі м’якого обчисленння – такі обчислювальні техніки підходять для виявлення мережевих аномалій, так як часто неможливо знайти точне рішення. М’які обчислення зазвичай включають такі методи, як генетичні алгоритми, штучні нейронні мережі, нечіткі множини, грубі множини, мурашині алгоритми та штучні імунні системи.

Генетичні алгоритми – це популяційні адаптивні евристичні методи пошуку, засновані на еволюційних ідеях. Такі методи спочатку перетворюють задачу в систему, яка використовує їромосову як структуру даних. На базі такого підходу булу розроблена генетична система виявлення вторгнень GBID, яка вивчає індивідуальну поведінку користувача. Поведінка користувача як набір із трьох елементів: індекс збігу, індекс ентропії, індекс новизни, та вивчається за допомогою генетичного алгоритму. Цей профіль поведінки використовується для виявлення вторгнень на основі попередньої поведінки.

Штучні нейронні мережі обумовлені визнанням того, що людський мозок обчислює інформацію зовсім інакше, ніж звичайний цифровий комп’ютер. Мозок структурує свої компоненти, відомі як нейрони, таким чином, щоб виконувати певні обчислення (наприклад, розпізнавання образів, сприйняття та управління рухом) у декілька разів швидше ніж найшвидший цифровий комп’ютер. Для досягнення гарної продуктивності, реальні нейронні мережі використовують массивні взаємозв’язки нейронів. Нейронні мережі набувають знання про навколишнє середовище через процес навчання, який систематично змінює

інтенсивні взаємозв'язки, або синаптичні ваги мережі для досягнення бажаної мети проектування.

Прикладом IDS на базі штучних нейронних мереж є система RT-UNNID. Така система здатна здатна забезпечити інтелектуальне виявлення вторгнень в режимі реального часу із використанням нейронних мереж навчених без вчителя. Архітектура мережі наведена на рисунку 2.10.

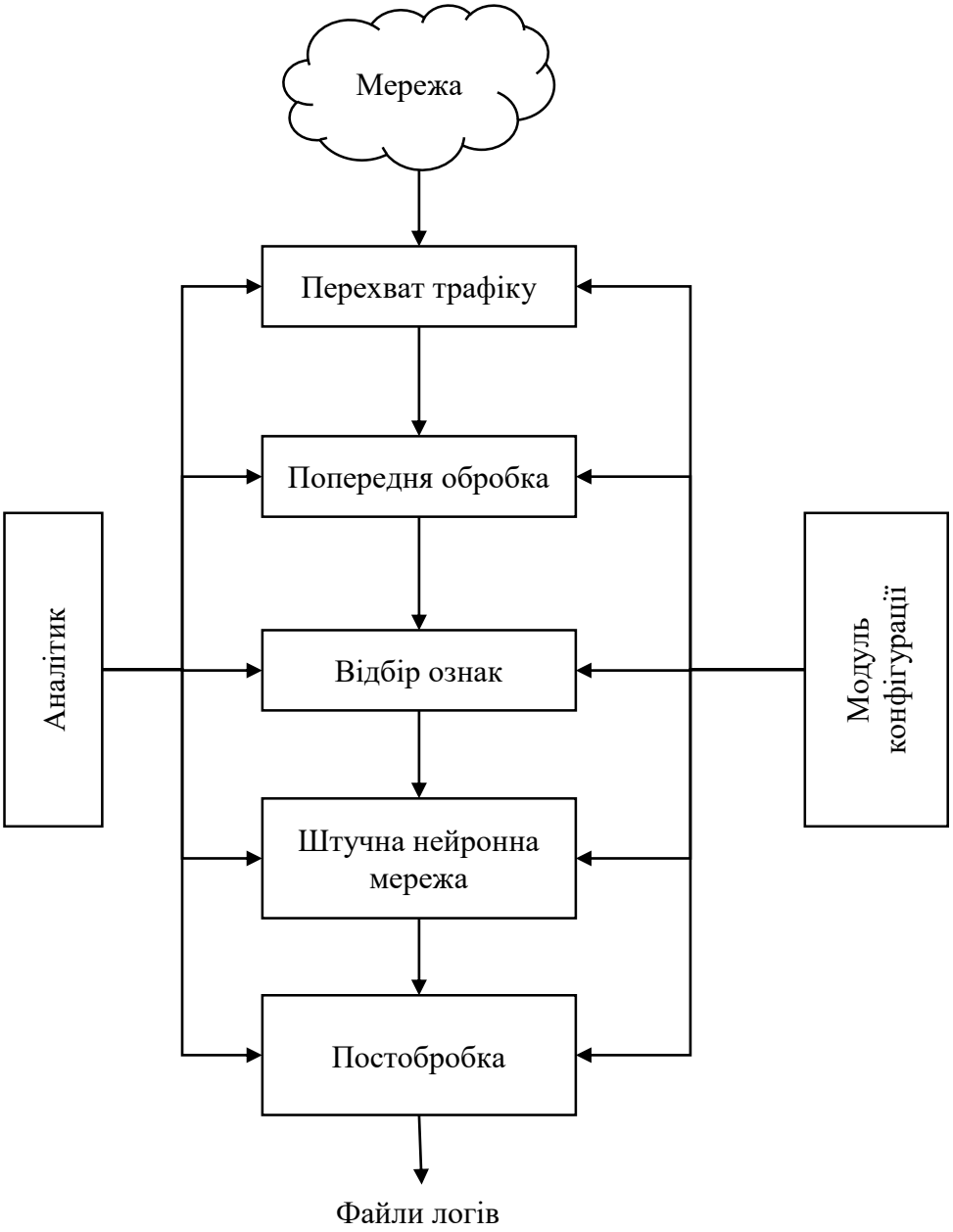

Рисунок 2.10 – Архітектура системи RT-UNNID

Спочатку системи перехвачує та обробляє дані мережевого трафіку для протоколів TPC, UDP ICMP. Модуль попередньоїх обробки витягує числові дані та перетворює їх в двійкову або нормалізовану форму. Перетворені дані посилаються в модуль виявлення на основі нейронної мережі, яка використовує адаптивну теорію резонанса та самоорганізаційну карту Кохонена. Далі, результат меанізму виявлення відправляється модулю постобробки для запису в файл системного журналу користувача та генерації сигналів тривоги при виявленні атаки. RT-UNNID може працювати в режимі реального часу для виявлення відомих та невідомих атак в мережевому трафіку із високою швидкістю виявлення.

Нечіткі множини використовуються у нечіткіх системах виявлення вторгнення у мережі. Такі системи використовують нечіткі правила для визначення ймовірності специфічних або загальних мережевих атак. Нечітка вхідна множина може бути визначена для трафіка в специфічній мережі.

FIRE (Fuzzy Intrusion Recognition Engine) – це система виявлення вторгнень на основі аномалій, використовуючи нечітку логіку для оцінки того, чи має місце шкідлива активність в мережі. Система поєднує прості метрики мережевого трафіку із нечіткіми правилами для виявлення ймовірності конкретних або загальних мережевих атак. Після того, як метрики стають доступними, вони оцінюються за допомогою теоретичного підходу із нечіткіми множинами. Система приймає нечіткі профілі мережевого трафіку в якості вхідних даних для свого набору правил та повідомляє про наявність загрози.

Оптимізація мурашиної колонії і зв’язані з нею алгоритми являються ймовірнісними методиками вирішення обчислювальних задач, які можуть бути переформульовані для знаходження оптимальних шляхів через графи. Алгоритми засновані на поведінці мурах, які шукають шлях між своєю колонією та джерелом їжі. На основі цих алгоритмів, було розроблено систему, як

використовує оптимізацію мурашиної колонії для виділення ознак для методу опорних векторів, який використовувався для виявлення мережевих вторгнень. Ознаки представлені у вигляді вузлів графу та ребрами між ними, які означають доповнення наступної ознаки. Мурахи проходять по графу для додавання нових вузлів до тих пір, поки не зустрінеться критерій зупинки.

Штучні імунні системи представляють собою обчислювальний метод, який заснований на принципах людської імунної системи, так як вона здатна виявляти аномалії. На базі штучної імунної системи було запропоновано систему, засновану на аналізі ефективності, для виявлення індивідуальної аномальної поведінки. Така система контродє мережу, аналізуючи набір параметрів для забезпечення загальної інформації про її стан. Для динамічної генерації стану системи використовується парадигма нечіткіх множин інтервального типу.

Переваги методів виявлення аномалій на базі м’яких обчислень:

- Такі системи навчання виявляють або класифікують функції постійні ознаки без зворотнього зв’язку із навколишнім середовищем.
- Завдяки адаптивній природі штучних нейронних мереж, можна навчати та тесутвати приклади даних інкрементно, використовуючи спеціальні алгоритми. Багатошарові нейромережеві методи більш ефективні, чим одношарові нейронні мережі.
- Навчання без вчителя із використанням конкуруючих нейронних мереж ефективно при кластеризації, виділення ознак та виявлення подібності.
- Грубі множини корисні для вирішення проблеми неузгодженості в наборі даних та для створення мінімального, не надмірного та не суперечливого набору даних.

Недоліки методів виявлення аномалій на базі м’яких обчислень:

- Під час навчання нейронних мереж може статись перенавчання.

- Якщо немає достовірних даних про нормальний трафік, то навчання таких методів стає дуже важким.
- Більшість методів мають проблеми із масштабованістю.
- Генерація правил на основі грубих множин страждає від доказу їх повноти.
- При використанні методів, заснованих на нечіткіх асоціаціях та на базі правил, виявлення відповідних підмножин правил та динамічне оновлення правил під час виконання являється складною задачею.

В методах на базі знань, події мережі або хоста перевіряються на відповідність із наперед визначеними правилами або шаблонами атак. Ціль методу заключається в тому, щоб узагальнити відомі атаки, щоб полегшити обробку реальних подій. Прикладами методів на базі знань є експертні системи, які базуються на правилах, онтології, логіці та аналізу переходу від одного стану до іншого. Такі методи шукають випадки відомих атак, намагаючись знайти збіг із заздалегідь заданими шаблонами атак. Пошук починається, як і інші методи виявлення вторгнень, при повній відсутності знань. Наступні збіги дій в мережі із відомою атакою допомагають покращити знання та х більшою впевненістю увійти в область. В кінці може бути показано, що подія або активність в мережі досягли максимального балу аномалії. Прикладом системи на базі знань є система STAT (State Transition Analysis Tool). Архітектура системи STAT наведена на рисунку 2.11.

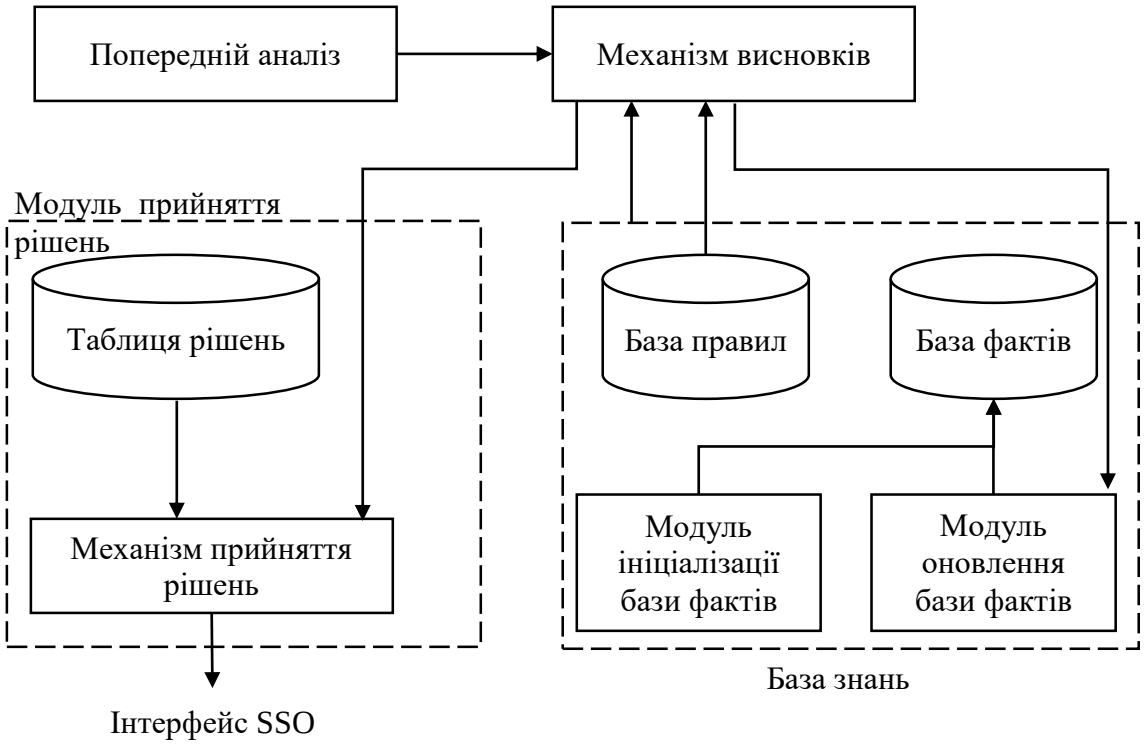

Рисунок 2.11 – Архітектура системи STAT

Система моделює дані трафіка як серію змін станів, які переходять із захищеного стану в цільовий скопроментований стан. STAT складається із трьох основних компонентів: бази знань, механізму висновків та механізму прийняття рішень. Попередня обробка даних обробляє та трансформує сирі дані для відправки в якості вхідних даних в механізм висновків. Механізм висновків відслідковує переходи станів, виокремлених із попередньо опрацьованих даних і потім порівнює ці стани із станами, доступними в базі знань. Механізм прийняття рішень контролює покращення механізму висновків для узгодження точності переходів станів. Він також визначає дії, які повинні бути зроблені на основі результатів механізму висновків та таблиці рішень. Нарешті, результати рішення відправляються в інтерфейс SSO для прийняття відповідних заходів. Таки алгоритми також здатні виявляти кооперативні атаки та атаки на протязі всіх користувацьких сеансів.

Переваги методів виявлення аномалій на базі знань:

- Ці методи надійні та гнучкі.
- Такі методи мають високий коефіцієнт виявлення, якщо вдасться належним чином зібрати обширну базу знань про атаки та про нормальну поведінку.

Недоліки методів на базі знань:

- Розробка високонадійних баз знань часто буває важкою та займає багато часу.
- Через відсутність прикладів атак та нормальної поведінки таки метод може генерувати велику кількість помилкових спрацьованих.
- Такі методи можуть бути не в стані виявити рідкісні або невідомі атаки.
- Динамічне оновлення бази знань або правил є складною справою.

Комбіновані методи та системи навчання. В цій секції буде розглянуто декілька методів та систем, які використовують комбінацію кількох методик, зазвичай класифікаторів.

Методи та системи на основі ансамблів. Ідея методології ансамблів полягає в тому, щоб оцінити декілька окремих класифікаторів та об’єднати їх для отримання загального класифікатора, який перевершує кожен із них окремо. Ці методи оцінюють індивідуальні позиції та об’єднують їх для прийняття кінцевого рішення. Методи на базі ансамблів поділяються на категорії в залежності від використовуваних підходів. Три основних підходів до створення ансамблів – це беггінг, бустинг та стакінг. Беттінг підвищує точність класифікації , створюючи покращений композитний класифікатор в єдиний проект, об’єднуючи результати вивчених класифікаторів. Бустинг створює ансамбль поступово, навчаючи неправильно класифіковані екземпляри, отримані із попередньої моделі. Стекінг досягає високої точності узагальнення за рахунок використання вихідних ймовірностей для кожної мітки класу із класифікаторів базового рівня.

Методи на базі ансамблю вигідні тим, що вони забезпечують більш високу точність порівняно із індивідуальними методами. Основні переваги таких методів:

- Навіть якщо окремі класифікатори слабкі, ансамблі методів добре працюють, комбінуючи декілька класифікаторів.
- Ансамбль методів може масштабуватись для великих наборів даних.
- Класифікатори ансамблів потребують набір керуючих параметрів, які являються комплексними та можуть бути легко налаштовані.
- Серед всіх існуючих методів, найбільш ефективними є Adaboost та стакінг, так як вони можуть використовувати різноманітність в прогнозуванні ха допомогою класифікаторів базового рівня.

Недоліки методів на основі ансамблів:

- Вибір підмножини послідовних працюючих та об’єктивних класифікаторів із усього набору класифікаторів є досить складним.
- Жадібний підхід до вибору наборів даних повільний для великих наборів даних.
- Важко домогтися продуктивності в режимі роботи в реальному часі.

Методи та системи на базі синтезу. Із ростом потреби автоматизованого прийняття рішень, важко підвищити точність класифікації порівняно їх окремими загальними методами на базі прийняття рішень, навіть якщо така системи може мати декілька непорівнянних джерел даних. Отже, підходяща комбінація із таки класифікаторів називається синтез. Декілька технік на базі синтезу було застосовано до виявлення мережевих аномалій. Класифікація таких методів виглядає наступним чином: рівень даних, рівень характеристик та рівень прийняття рішень. Деякі методи вирішують проблему роботи тільки в просторі високої розмірності із ознаками, які розділені на семантичні групи. Інші

намагаються поєднати класифікатори, підготовлені по різним ознакам, які поділяються по ієрархічним рівням абстракції або типу інформації.

HMMPayl є прикладом системи виявлення вторгнень на базі синтезу, де корисне навантаження відображається як послідовність байтів які аналізуються з використанням прихованих марківьких моделей. Алгоритм відбирає ознаки та використовує приховані марківські моделі для забезпечення тієї ж обчислювальної здатності, що і аналіз n-gram, долаючи при цьому його обчислювальну складність. HMMPayl дотримується парадигми системи множинних класифікаторів, щоб забезпечити кращу точність класифікації, збільшити складність ухилення від IDS, та мінімізувати слабкі місця, пов’язані із неоптимальним вибором параметрів для прихованої марківської моделі.

Переваги методів на базі синтезу:

- Синтез даних ефективний для підвищення своєчасності ідентифікації атаки та скорочення числа хибних відпрацювань.
- Синтез відповідних навчальних даних та рівня прийняття рішень зазвичай призводить до високої швидкості виявлення.

Недоліки:

- Великі обчислювальні затрати для ретельної підготовки зразків.
- Синтез рівня ознак являється доволі складною та тривалою задачею. Крім того, зміщення між базовими класифікаторами впливає на процес синтезу.
- Складність побудови гіпотези для різних класифікаторів.

Гібридні методи та системи. Більшість систем виявлення мережевих вторгнень використовують сигнатурне порівняння виявлення аномалій. Однак, сигнатурне порівняння не може виявити невідомі вторгнення, а аналіз аномалій зазвичай має високий відсоток хибно позитивних результатів. Для подолання обмежень таких методів, розробляються гібридні методи, які використовують особливості декількох підходів до виявлення мережевих аномалій. Як приклад

таких систем можна розглянути модель FLIPS (Feedback Learning IPS), яка використовується для захисту хоста та запобігає атаки їх ін'єкцією двійкового коду. Така система включає в собі три основних компоненти: класифікатор на базі аномалій,схему фільтрації на базі сигнатур та систему контролю, яка використовує ISR. Перехват двійкового коду дозволяє моделі FLIPS створювати сигнатури для експлойтів нульвого дня. Також був запропонований підхід, який поєднує дерева рішень та метод опорних векторів в якості ієрархічної гібридної моделі інтелектуальної системи для виявлення вторгнень. Такий підхід максимізує точність виявлення та мінімізує складність обчислення.

Переваги гібридних методів:

- Таки метод використовує основі особливості сигнатурного та аномального методів виявлення мережевої аномалії.
- Такі методи можуть запобігти як відомим так і не відомим атакам.

Недоліки:

- Відсутність належної гібридизації може привести до високих обчислювальних витрат.
- Динамічні зміни правила, профілю або сигнатури залишається все ще складною задачею.

Висновки до розділу

В цьому розділі було розглянуто методи виявлення вторгнень та системи виявлення мережевих вторгнень на базі аномалій поділених на декілька категорій. Після аналізу методів, можна зробити декілька спостережень:

- Кожен клас методів та систем виявлення мережевих вторгнень на базі аномалій має слабкі та сильні сторони. Придатність того чи іншого методу виявлення вторгнень залежить від характеру проблеми, яку потрібно вирішити. Відтак, забезпечення єдиного інтегрованого рішення для кожної проблеми виявлення аномалій може бути недоцільним.
- При використанні складних наборів даних різні методи стикаються із різноманітними складнощами. Метод найближчих сусідів та кластеризації страждають коли дані мають велику розмірність, оскільки міри відстаней не здатні адекватно розрізнити нормальні та аномальні випадки використовуючи дані із великими розмірностями. Спектральні методи явним чином вирішують проблему великої розмірності шляхом зіставлення даних з більш низькою розмірною проекцією. Але їх продуктивність сильно залежить від припущення, що приклади даних нормальної та аномальної поведінки помітні в проектованому просторі. При такому підході, класифікація часто працює краще. Проте, вона вимагає розмічених навчальних даних як для прикладів нормальної поведінки так і для прикладів атак. Неправильна розмітка навчальних даних часто робить ускладнює задачу навчання.
- Для виявлення вторгнень в режимі реального часу, складність процесу виявлення аномалій грає важливу роль. У випадку класифікації, кластеризації та статистичних методів, хоча навчання є витратним, вони все ж є прийнятними, тому що тестування є швидким, а навчання відбувається в режимі оффлайн.
- Методи виявлення аномалій зазвичай припускають, що аномалії в даних зустрічаються рідко в порівнянні зі звичайними випадками. Як правило, такі припущення вірні, але не завжди. Часто методи навчання без учителя страждають від великої кількості помилкових спрацьовувань, коли

аномалії виникають у великих кількостях. Для виявлення масових аномалій можуть застосовуватися методи, що були навчені з учителем або в напівавтоматичному режимі.

Після аналітичного огляду переваг наявних методів, для подальшої роботи було обрано декілька методів, які накраще підходять до теми даної роботи.

РОЗДІЛ 3 ПРОЕКТУВАННЯ ТА РОЗРОБКА МОДЕЛІ

3.1 Методи та засоби

Python — це вільна об’єктно орієнтована мова програмування, яка приваблює простим синтаксисом та динамічною структурою. В python дуже просто писати та аналізувати код. Іншою перевагою є наявність дуже великого об’єму навчально літератури та відеоматеріалів (книги, сайти, форми і т.д.). В доповненні до наведених переваг, python працює стабільно із багатьма бібліотеками , за допомогою яких можна імплементувати машинне навчання. Для виконання проекту було обрано версію останню версію python 3.6.

Sklearn (Scikit-learn) — це безкоштовна програмна бібліотека машинного навчання для мови програмування Python, яка надає функціональність для створення та тренування різноманітних алгоритмів класифікації, регресії та кластеризації, таких як лінійна регресія, random forest, градієнтний бустинг, і працює у зв'язці з бібліотеками NumPy та SciPy. Scikit-learn є однією з найбільш популярних бібліотек машинного навчання. Sklearn має дуже багату документацію та містить всі алгоритми потрібні для виконання даної роботи.

Pandas — програмна бібліотека, написана для мови програмування Python для маніпулювання даними та їхнього аналізу. Вона, зокрема, пропонує структури даних та операції для маніпулювання чисельними таблицями та часовими рядами.

Matplotlib — бібліотека на мові програмування Python для візуалізації даних двовимірною 2D графікою. Matplotlib є гнучким, легко конфігурованим пакетом, який разом з NumPy, SciPy і IPython надає можливості, подібні до MATLAB. В даний час пакет працює з декількома графічними бібліотеками, включаючи wxWindows і PyGTK.

Numpy — розширення мови Python, що додає підтримку великих багатовимірних масивів і матриць, разом з великою бібліотекою високорівневих математичних функцій для операцій з цими масивами. NumPy можна розглядати як гарну вільну альтернативу MATLAB, оскільки мова програмування MATLAB зовні нагадує NumPy: обидві вони інтерпретовані, і обидві дозволяють користувачам писати швидкі програми поки більшість операцій проводяться над масивами або матрицями, а не над скалярами.

Критерієм для оцінки алгоритмів машинного навчання є час виконання. Однак, тривалість виконання може залежати від апаратних потужностей обчислювальної машини, на якій виконуються вирахування. В зв’язку з цим, наведено характеристики комп’ютера, на якому запускалось зазначене вище програмне забезпечення:

CPU: Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz 2.90 GHz

RAM: 8GB

OS: WINDOWS 10 Pro 64-bit

GPU: NVIDIA GEFORCE 1060TI

Результуюча система буде оцінюватись по чотирьом критеріям: accuracy, precision, f-measure, recal. Всі ці критерії мають значення від 0 до 1 - коли критерій наближається до 1 то продуктивність збільшується, а коли наближається до 0 то продуктивність системи зменшується.

Accuracy - співвідношення успішно категоризованих даних до всіх даних (формула 3.1)

$$Accuracy = \frac{TN + TP}{FP + TN + TP + FN} \tag{3.1}$$

Recall – відношення даних, класифікованих як атака, до всіх даних атаки. (формула 3.2)

$$Recall = \frac{TP}{TP + FN} \tag{3.2}$$

Precision - це співвідношення успішно позитивно класифікованих зразків до всіх позитивно класифікованих зразків (формула 3.3)

$$Precision = \frac{TP}{FP + TP} \tag{3.3}$$

F-measure - це середнє гармонійне від precision та recall (формула 3.4)

$$F - measure = \frac{2}{\frac{1}{Recall} + \frac{1}{Precision}} \tag{3.4}$$

F-measure використовують для вираження загального успіху, відповідно, в цій роботі при аналізі різних підходів головний акцент буде надано саме на цю матрику.

При обчисленні цих метрик використовуэться чотири значення, наведених нище:

- TP – True Positive (Істинно позитивний). Зразки атаки класифіковані як атаки.
- FP – False Positive (Помилка першого роду). Доброякісні зразки класифіковані як атаки.
- FN – False Negative (Помилка другого роду). Зразки атаки класифіковано як доброякісні.
- TN – True Negative (Істинно негативний). Доброякісні зразки класифіковані як доброякісні.

Це розподілення представлене за допомогою матриці невідповідності (англ. confusion matrix) на рисунку 3.1.

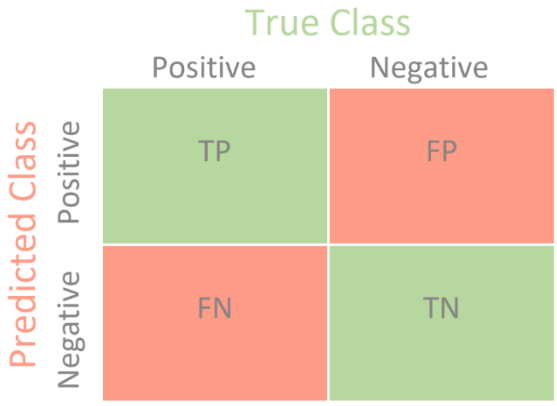

Рисунок 3.1 – Матриця невідповідності

В доповненні до наведених вище чотирьох критеріях, оцінка моделей також буде проводитись по тривалості обчислення, що є також доволі важливим фактором, хоч і не критичним.

3.2 Попередня підготовка набору даних

В цій секції проводяться попередня обробки даних для використання їх у методах машинного навчання для виявлення аномалій. Для цього спочатку було зроблено очистку навчального набору від помилкових та дефективних прикладів. Далі було поділено навчальну вибірку на дві частини - тренувальну та тестову вибіркі. Наступний крок полягає у відбору ознак - процес відбору підмножини доречних ознак для використання в побудові моделі.

Перед використанням датасету, може бути необхідним внести деякі корективи до набору даних. Для цього в даній секції виправлені деякі дефекти датасету CICIDS2017, а також відредаговані деякі дані.

Файл датасету містить 3119345 потокових записів. Розподілення цих записів наведені в таблиці 3.1 та на рисунку 3.2

Таблиця 3.1 – Розподілення даних в датасеті CICIDS2017

Мітка	Кількість
Benign	2359289
Faulty	288602
DoS Hulk	231073
PortScan	158930
DDoS	41835
DoS GoldenEye	10293
FTP-Patator	7938
SSH-Patator	5897
DoS slowloris	5796
DoS Slowhttptest	5499
Bot	1966
Web Attack –Brute Force	1507
Web Attack –XSS	652
Infiltration	36
Web Attack –SQLInjection	21
Heartbleed	11

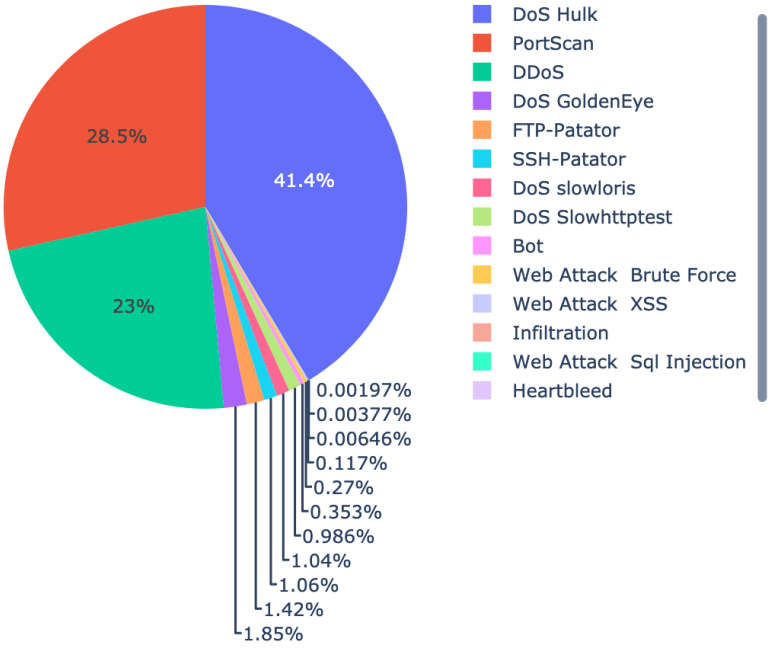

Рисунок 3.2 – Розподілення даних в датасеті CICIDS2017

Після аналізу цих записів, видно що 288602 записів є невірними/неповними. Перший крок попередньої обробки буде видалення цих непотрібних записів.

Також, після початкового аналізу було виявлено іншу неточність в колонках, з яких складаються ознаки. Датасет складається із 86 колонок, які визначають характеристики потоку, такі як Flows ID, Source IP, Source Port. Однак, колонка Fwd Header Length була записана два рази (41 колонка та 62 колонка). Цю неточність була виправлена видаленням повторюваної колонки.

Ще одним необхідним кроком в попередній обробці даних є перетворення ознак, які мають категорійні та строкові значення (Flow ID, Source IP, Destination IP, Timestamp, External IP) в числові значення, для того щоб можна було використовувати ці ознаки в алгоритмах машинного навчання. Таке перетворення може бути здійсненим за допомогою класу LabelEncoder із

бібліотеки Sklearn. Таким чином, різні строкові значення, які не можуть бути використані в алгоритмах машинного навчання, отримають числові значення в діапазоні від 0 до n-1, і стануть більш підходящими для опрацювання. Варто зазначити, що в датасеті також присутня колонка з строковими значеннями - Label, яка не зазнала ніяких змін через те, що для подальшої обробки потрібні оригінальні значення колонки для успішної класифікації різних видів атак.

Зрештою, також необхідно виконати наступні незначні коректування в датасеті:

- В колонці Label, символ "–"(код юнікоду &#8211), який використовується для ідентифікації підтипів веб атак, не розпізнається кодеком utf-8, тому повинен бути замінений на символ "-" (код юнікоду &#45).
- Колонки Flow Bytes/s, Flow Packets/s містять значення "Infinity" та "NaN", які можуть бути конвертовані в -1 та 0 відповідно, щоб алгоритми машинного навчання могли опрацювати такі значення.

Навчальний набір необхідний для успішної роботи алгоритмів машинного навчання. Тестувальний набір потрібен для кінцевої оцінки алгоритму та перевірці адекватності роботи моделі на невідомих даних. Однак, датасет CICISD2017, який використовується в поточній роботі, не містить спеціальних даних для навчання та тестування, тільки єдиний розмічений набір даних. Тому потрібно вручну розділити датасет на навчальний та випробувальній набір відповідно. Для цього використовується функція train_test_split із бібліотеки Sklearn, яка розділяє потрібний набір даних на дві частини з розміром, вказаним користувачем. Як правило, кращим співвідношення поділу є 20% даних для випробувального набору та 80% даних для навчального набору відповідно. Така ж сама пропорція використовується і в даному проекті.

Функція train_test_split випадково вибирає дані при створені наборів. Такий процес називається крос валідацією (cross-validation). Для забезпечення

достовірності результатів отриманих під час поділу, алгоритм поділу запускався 10 разів поспіль. Отримані результати представляють собою середнє арифметичне повторюваних операцій.

Для оптимізації роботи алгоритмів машинного навчання потрібно проаналізувати всі доступні ознаки набору даних та які саме ознаки найбільш важливі для кожного із типів наявних атак.

Для обчислення важливості ознак для кожного типу атак, було створено набори даних, які складаються із доброякісних прикладів та із прикладів атак тільки одного типу. Розподілення позитивних та негативних прикладів відбувалось у відношенні 30% на приклади атак та 70% на доброякісні приклади.

Важливість ознак обчислювалась за допомогою алгоритму Random Forest Regressor. Цей алгоритм створює дерева прийняття рішень, в яких кожній із ознак придається вага значущості з точки зору того, наскільки ця ознака корисна при побудові дерева прийняття рішень. Після закінчення роботи алгоритму, ваги значущості кожної ознаки порівнюються та сортуються. Сума всіх вагів значущості всіх характеристик дає загальну вагу дерева прийняття рішень. Порівняння оцінки будь-якої ознаки із значенням оцінки всього дерева дає інформацію про те, наскільки важлива ця ознака в дереві прийняття рішень.

Однак, вісім ознак (Flow ID, Source IP, Source Port, Destination IP, Destination Port, Timestamp, External IP) мають бути виключені із обчислень ваг значущості. Хоча ці ознаки використовуються в класичному підході, не виключено, що зловмисник віддасть перевагу не використовувати відомі порти для проникнення в систему або обходу обмежень операційної системи, або він може використовувати згенеровані/підроблені IP - адреси. Також, багато портів використовуються динамічно, та багато програм використовують одні і ті ж порти. Так що, використання номеру порту може значно спотворити кінцевий результат моделей.

Отже, для ефективного відбору ознак краще прибрати ті ознаки, які є не сталими та не узагальненими (IP address, port number, timestamp). Натомість, варто використовувати більш загальні та інваріантні атрибути для визначення атаки. Тому що форма даних надать набагато більше інформації про те, чи є це атакою чи ні.

Розподіл атак та чотирьох найбільш значущих ознак для кожної із атак наведено в таблиці 3.2.

Таблиця 3.2 – Розподіл атак та чотирьох найбільш значущих ознак для кожної атаки

Тип атаки	Ознака	Вага ознаки
SSH-Patator	Flow Bytes/s	0.000846
	Total Length of Fwd Packets	0.000814
	Fwd Packet Length Max	0.000749
	Flow IAT Mean	0.000734
DoS Hulk	Bwd Packet Length Std	0.514306
	Fwd Packet Length Std	0.069838
	Fwd Packet Length Max	0.008542
	Flow IAT Min	0.001716
FTP-Patator	Fwd Packet Length Max	0.063671
	Fwd Packet Length Std	0.022751
	Fwd Packet Length Mean	0.002179

Продовження таблиці 3.2

Тип атаки	Ознака	Вага ознаки
	Total Length of Bwd Packets	0.000746
DoS GoldenEye	Flow IAT Max	0.442727
	Bwd Packet Length Std	0.091185
	Flow IAT Min	0.053795
	Total Backward Packets	0.041583
Bot	Bwd Packet Length Mean	0.304823
	Flow IAT Max	0.034495
	Flow IAT Std	0.019464
	Flow Duration	0.010129
Infiltration	Total Length of Fwd Packets	0.05238
	Flow IAT Max	0.036096
	Flow Duration	0.016453
	Flow IAT Min	0.015448
Web Attack	Total Length of Fwd Packets	0.014697
	Bwd Packet Length Std	0.00536
	Flow Bytes/s	0.00257
	Bwd Packet Length Max	0.001922

Продовження таблиці 3.2

Тип атаки	Ознака	Вага ознаки
DDoS	Bwd Packet Length Std	0.468089
	Total Backward Packets	0.094926
	Fwd IAT Total	0.012066
	Total Length of Fwd Packets	0.006438
Heartbleed	Bwd Packet Length Mean	0.064
	Total Length of Bwd Packets	0.056
	Flow IAT Min	0.056
	Bwd Packet Length Std	0.044
PortScan	Flow Bytes/s	0.313402
	Total Length of Fwd Packets	0.304917
	Flow Duration	0.000485
	Fwd Packet Length Max	0.00013
DoS Slowhttptest	Flow IAT Mean	0.64206
	Fwd Packet Length Min	0.075942
	Fwd Packet Length Std	0.022194
	Bwd Packet Length Mean	0.020857
DoS slowloris	Flow IAT Mean	0.465561

Кінець таблиці 3.2

Тип атаки	Ознака	Вага ознаки
	Bwd Packet Length Mean	0.075633
	Total Length of Bwd Packets	0.049808
	Total Fwd Packets	0.01868

Після аналізу важливих ознак для кожної атаки, видно що одна або дві ознаки присутні майже для всіх типів атак. З іншої сторони, характеристика атак Heartbleed та SSH-Patator виглядає трохи незвичайно. Для цих атак декілька значення важливості ознак дуже близькі один до одного, тобто немає однієї домінантної ознаки.

Також, для атаки PortScan, дві ознаки Flow Bytes/s та Total Length of Fwd Packets виділяються серед інших. Суть цієї атаки полягає в тому, що зловмисник відправляє настільки багато пакетів без пейлоаду наскільки можливо (тільки 20 байтів для заголовків транспортного рівня та 20 байтів для заголовків інтернет рівня, 40 байтів загалом) або з мінімальним пейлоадом. Таким чином, атакуючий як і прискорює процес так і робить атаку більш ефективно та не займає всю пропускну здатність мережевого шлюзу без необхідності.

В результаті аналізу, очікується що ознака Total Length of Fwd Packets повинна мати дуже малу значущість в порівнянні із доброякісними потоками. Дані, зображені на рисунку 3.3, підтверджують цей прогноз.

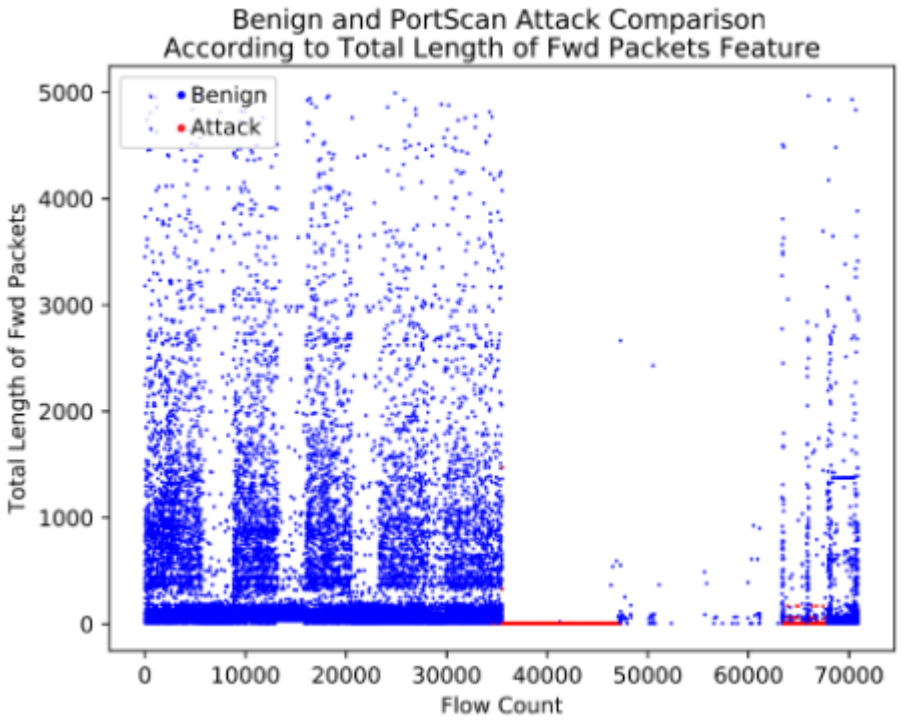

Рисунок 3.3 – Порівняння значень ознаки Total Length of Fwd Packets в прикладах нормальної поведінки та прикладів атаки PortScan

Натомість, аналіз атаки SSH-Patator не показав ніяких очевидних головних ознак. Значення значущості для кожної ознаки дуже близькі один до одного. Найбільш важлива причина для цього полягає в тому, що атака SSH-Patator не складається із одного кроку, це складна структура з трьома етапами (Сканування, Brute Force та фаза Die Of). Дана атака містить як і Portscan атаку так і Brute Force атаку. Порівняння ознаки в прикладах з атакою та доброякісними прикладами можна побачити на рисунку 3.4.

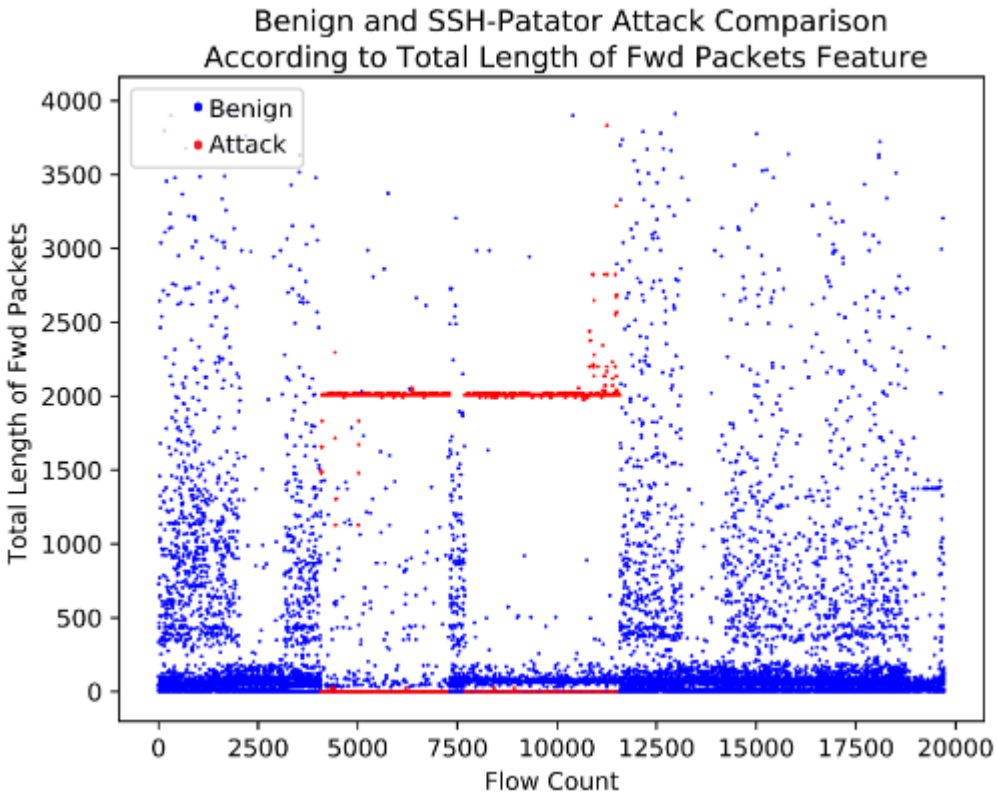

Рисунок 3.4 – Порівняння значень ознаки Total Length of Fwd Packets в прикладах нормальної поведінки та в прикладах атаки SSH-Patator

Інший підхід в відборі ознак це застосування алгоритму Random Forest Regressor до всього датасету, збираючи всі типи атак під однією міткою. Таким чином, дані в датасті будуть тільки мітки “Attack” або “Benign”(Доброякісна). В результаті опрацювання датасету, було отримано список найважливіших ознак, наведених в таблиці 3.3.

Таблиця 3.3 – Ваги ознак

Назва ознаки	Вага	Назва ознаки	Вага
Bwd Packet Length Std	0.246627	Flow IAT Mean	0.003266
Flow Bytes/s	0.178777	Total Length of Bwd Packets	0.001305
Total Length of Fwd Packets	0.102417	Fwd Packet Length Min	0.000670

Кінець таблиці 3.3

Назва ознаки	Вага	Назва ознаки	Вага
Fwd Packet Length Std	0.063889	Bwd Packet Length Mean	0.000582
Flow IAT Std	0.009898	Flow Packets/s	0.000541
Flow IAT Min	0.006946	Fwd Packet Length Mean	0.000526
Fwd IAT Total	0.005121	Total Backward Packets	0.000169
Flow Duration	0.004150	Total Fwd Packets	0.000138
Bwd Packet Length Max	0.004007	Fwd Packet Length Max	0.000125
Flow IAT Max	0.003579	Bwd Packet Length Min	0.000084

Два різних підходи було обрано для застосування алгоритмів машинного навчання.

Перший підхід, це використання наборів даних тільки з один типом атак та доброякісних прикладів, та застосування ознак які найбільш важливі для для цього типу атаки. Ці набори даних містять 30% прикладів атаки та 70% доброякісних прикладів. Далі, до кожного набору даних було застосовано сім різних алгоритмів машинного навчання. Таким чином, можна буде спостерігати за ефективністю та продуктивністю роботи різних методів машинного навчання на різних типах атак.

Другий підхід це, використовується весь набір даних. Всі атаки в цьому наборі позначені однією загальною міткою “attack”, в результаті чого, набір даних містить тільки приклад атаки або доброякісний приклад. Відбір ознак для цього підходу полягає в об’єднанні всіх 4 найголовніших ознак для 12 видів атаки. В результаті було отримано 48 ознак, та після видалення повторень, було

отримано 18 ознак. Список фінальних найголовніших ознак наведено в таблиці 3.4.

Таблиця 3.4 – Список найважливіших ознак для всіх типів атак

Bwd Packet Length Max	Flow IAT Mean	Fwd Packet Length Min
Bwd Packet Length Mean	Flow IAT Min	Fwd Packet Length Std
Bwd Packet Length Std	Flow IAT Std	Total Backward Packets
Flow Bytes/s	Fwd IAT Total	Total FwdPackets
Flow Duration	Fwd Packet Length Max	Total Length of Bwd Packets
Flow IAT Max	Fwd Packet Length Mean	Total Length of Fwd Packets

Як альтернативний шлях відбору ознак є вибір ознак із найбільшою вагою отриманих в результаті опрацювання всього набору даних. Для відбору ознак було встановлено поріг значення в 0.8%. Таким чином, 97% від загальних ваг значущості будуть покриті всього лише 7 ознаками. Решта 18 ознак становлять тільки 3% від загальних ваг значущостей. В результаті порівняння ваги ознаки з встановленим порогом, було набір ознак, які наведені в таблиці 3.5.

Таблиця 3.5 – Список відібраних ознак в результаті опрацювання всього набору даних

Назва ознаки	Вага	Відсоток
Bwd Packet Length Std	0.246627	38.97%
Flow Bytes/s	0.178777	28.25%
Total Length of Fwd Packets	0.102417	16.18%
Fwd Packet Length Std	0.063889	10.10%

Кінець таблиці 3.5

Назва ознаки	Вага	Відсоток
Flow IAT Std	0.009898	1.56%
Flow IAT Min	0.006946	1.10%
Fwd IAT Total	0.005121	0.8 %

3.3 Побудова моделей

В цьому підрозділі представлені результати досліджень та аналізу інформаційного забезпечення. При проведенні оцінки моделей найбільша увага буде звертатись на критерій F-measure. Тим не менше, всі отримані критерії будуть наведені у відповідних таблицях. Процедура оцінки ефективності буде проведена 10 разів для кожного із алгоритмів машинного навчання. Числа, які будуть представлені в таблицях, являються середнім арифметичним усіх десяти повторень.

Також, в цьому розділі повністю весь набір даних буде використовуватись для навчання моделей. Всі атаки в наборі даних мають одну загальну мітку “attack”. Сім різних методів машинного навчання були застосовані до поточного набору даних:Naive Bayes, QDA, Random Forest, ID3, XGBoost, MLP, KNN. Для побудови моделей буде використовуватись два підходи з використанням ознак: об’єднання найважливіших ознак кожної атаки та ознаки отримані в результаті опрацюванні всього набору даних.

Після обчислення ознак для кожної із різних типів атак, було побудовано 7 моделей. Оцінка результату кожної із моделей наведено в таблиці 3.6.

Таблиця 3.6 – Оцінка результату моделей побудованих за допомогою 18 ознак

Алгоритм машинного навчання	Критерії оцінки				
	F-measure	Precision	Recall	Accuracy	Time(seconds)
KNN	0.95	0.95	0.95	0.95	391.804
Random Forest	0.94	0.95	0.94	0.94	24.739
XGBoost	0.96	0.96	0.97	0.97	1967.054
ID3	0.95	0.95	0.95	0.95	29.284
QDA	0.30	0.84	0.31	0.31	6.649
MLP	0.79	0.81	0.84	0.84	81.668
Naive Bayes	0.79	0.80	0.78	0.78	4.576

Після аналізу результатів, видно що найкращий результат має алгоритм XGBoost із значенням 0.96. Наступні значення, 0.95, мають алгоритми ID3 та KNN. Однак, алгоритм ID3 працює швидше чим алгоритм XGBoost, так що ці алгоритми можуть розглядатись як ті, які мають найкращі показники. Найгірший показник має алгоритм QDA із значенням 0.3. Значення QDA майже на 0.4 пункту нижче ніж алгоритми (Naive Bayes та MLP) із найближчими значеннями.

З точки зору швидкості, найшвидшими алгоритмами є NB та QDA. Незважаючи на те, що XGBoost має найкращий результат, він набагато порядків повільніше порівняно з іншими алгоритмами.

Далі були побудовані моделі за допомогою загальних ознак, відібраних після обчислення всього набору даних. Результати побудови 7 різних моделей наведено в таблиці 3.7.

Таблиця 3.7 – Результати моделей побудованих за допомогою загальних ознак

Алгоритми машинного навчання	Критерії оцінки				
	F-measure	Precision	Recall	Accuracy	Time(seconds)
KNN	0.95	0.95	0.95	0.95	144
Random Forest	0.94	0.95	0.94	0.94	20
XGBoost	0.97	0.96	0.97	0.97	1038
ID3	0.95	0.95	0.95	0.95	11
QDA	0.41	0.84	0.31	0.31	1.9
MLP	0.79	0.81	0.84	0.84	51.7
Naive Bayes	0.81	0.80	0.78	0.78	1.6

Значення в таблиці 3.7, які відрізняються від аналогічних в таблиці 3.6, додатково підкреслені. Після порівняння цих двох таблиць, видно що метрика F-measure для алгоритмів Random Forest, ID3, XGBoost та MLP не зазнала ніяких змін. Однак, для алгоритмів Naive Bayes та QDA було помічено значне збільшення значення показника F-measure на 2 та 11 пунктів відповідно.

З точки зору швидкості, час виконання для всіх алгоритмів значно скоротився. Причина такого скорочення часу виконання є те що в першому випадку застосовувались 18 ознак, натомість в другому випадку тільки 7 ознак. Це зменшення ознак також зменшило час виконання всіх алгоритмів машинного алгоритму.

Висновок до розділу

В даному розділі було наведено методи та засоби, які необхідні для проведення експерименту із побудови моделей для виявлення найкращої моделі. Було наведено критерії по яким оцінювались побудовані моделі. Спочатку, навчальний датасет було оброблено та перевірено на рахунок невалідних записів, які в подальшому могли потенційно спотворити кінцевий результати досліджуваних моделей.

Так як навчальний датасет складався із великої кількості ознак, було застосовано два різних підходи до вибору ознак із навчальних даних – на основі обчислення всього датасету, де всі приклади атак були помічені однією міткою, та на основі обчислення кожної атаки окремо. На основі відібраних ознак було навчено декілька моделей в результаті чого були отримані наступні метрики F-measure: Naive Bayes - 0.86, QDA - 0.86, Random Forest - 0.94, ID3 - 0.95, K Nearest Neighbours - 0.94, MLP - 0.83, XGBoost - 0.97. Аналізуючи отримані значення метрики, можна зробити висновок, що найкращий результат має модель XGBoost.

РОЗДІЛ 4 МАРКЕТИНГОВИЙ АНАЛІЗ СТАРТАП-ПРОЕКТУ

4.1 Опис ідеї проекту

У цьому розділі описано стійке економічне обґрунтування можливості реалізації та імплементації стартапу “Система виявлення мережевих аномалій”. Запропонована технологія буде імплементована у вигляді програми та буде надаватися користувачам у формі покупки.

Розділ включає:

а) імплементацію технології;

б) розробку стратегії виходу на ринок та розвиток проекту.

У таблиці 4.1 розглянута основна ідея та напрямки використання продукту. Також описана цінність яку має запропонований продукт. Реалізація продукту буде базуватись на виявленні аномалій у мережі користувача. Компанії та спеціалісти кібербезпеки отримають змогу швидко та якісно виявляти вторгнення в їхні мережі.

Таблиця 4.1 – Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Ідея полягає в продажі системи IDS, яка на основі визначення мережевих аномалій буде виявляти вторгнення у мережу	Управління інформаційною безпекою	Виявлення аномалій в мережі; Моніторинг, візуалізація та аналіз швидкодії масового збору даних з багатьох внутрішніх корпоративних та зовнішніх джерел; Надання звітів по безпеці мережі.

Кінець таблиці 4.1

Зміст ідеї	Напрямки застосування	Вигоди для користувача
	Управління обчислювальними ресурсами	Оптимізація часу аналізу мережевого потоку; Інтелектуальне управління ресурсами системи

В таблиці 4.2 розглянуті слабкі та сильні сторони майбутнього продукту, це беззаперечно є основною складовою його конкурентноздатності. Сильною стороною є використання штучного машинного навчання для виявлення вторгнення в мережу. Слабкою стороною є обмежений функціонал в порівнянні із доступними конкурентами.

Таблиця 4.2 – Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п/п	Техніко-економічні характеристики ідеї	(Потенційні) товари/концепції конкурентів				W(слабка сторона)	N(нейтральна сторона)	S(сильна сторона)
		Мій проект	McAfee NSP	Hillstone NIPS	Huawei NIP			
1.	Вартість експлуатації	1500$	10995$	$18000	2000$	-	-	+
2.	Вартість обслуговування	Безкоштовно	Безкоштовно	180$	250$/міс.	-	-	+
3.	Наявність шифрування	Присутня	Присутня	Відсутня	Відсутня	-	+	-

Кінець таблиці 4.2

№ п/п	Техніко-економічні характеристики ідеї	(Потенційні) товари/концепції конкурентів				W(слабка сторона)	N(нейтральна сторона)	S(сильна сторона)
		Мій проект	McAfee NSP	Hillstone NIPS	Huawei NIP			
4.	Інтелектуальне балансування навантаження	Присутнє	Присутнє	Присутнє	Присутнє	+	-	-
5.	Забезпечення високої доступності	Присутнє	Відсутнє	Присутнє	Присутнє	-	+	-
6.	Підтримка Docker	Присутня	Відсутня	Присутня	Присутня	-	-	+
7.	Наявність консольного інтерфейсу	Присутній	Присутній	Відсутній	Відсутній	-	-	+

4.2 Технологічний аудит ідеї проекту

У даному підрозділі наводяться технології за допомогою яких була створена система виявлення мережевих аномалій. Технологічну здійсненність проекту наведено в таблиці 4.3.

Таблиця 4.3 – Технологічна здійсненність проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1.	Мова програмної реалізації	Компільована у машинний код (C, C++)	Наявні	Доступні
2.		Компільована у байткод (Java, C#)	Наявні	Доступні
3.		Скриптова/інтерпретована (Python, Perl, Lua, Ruby)	Наявні	Доступні
Обрана технологія реалізації ідеї проекту: Скриптова/інтерпретована (Python)				
4.	Data platform	Cloudera	Наявні	Недоступні
5.		Hortonworks	Наявні	Доступні
6.		Syncfusion	Наявні	Доступні
Обрана технологія реалізації ідеї проекту: Syncfusion.				

Розробку системи вирішено виконати на мові програмування Python та із використанням Syncfusion в якості хмарного сховища.

4.3 Аналіз ринкових можливостей запуску стартап-проекту

Аналіз ринкових можливостей необхідних для того щоб визначити в якому є його динаміка, чи існують якісь перешкоди та чи є конкурентні продукти. Також, можна буде змоделювати та запланувати подальший ринок стартап-

проекту. В таблиці 4.4 наведено попередню характеристику потенційного ринку стартап-проекту.

Таблиця 4.4 – Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1.	Кількість головних гравців, од	5
2.	Загальний обсяг продаж, дол/ум.од	1500$/ум. од.
3.	Динаміка ринку (якісна оцінка)	Зростає
4.	Наявність обмежень для входу (вказати характер обмежень)	Недискримінаційні якісні
5.	Специфічні вимоги до стандартизації та сертифікації	Відсутні
6.	Середня норма рентабельності в галузі (або по ринку %)	70%

Рентабельність галузі за потенційними показниками є більшою за прибуток від вкладання грошей до банку, тому стартап-проект є досить привабливим для інвесторів. Необхідно зазначити, що на даний момент на ринку немає обмежень для виходу стартап-проекту на ринок та якихось перешкод таких як проходження специфічної сертифікації.

В таблиці 4.5 обрані можливі групи потенційних клієнтів та переваг, які надає продукт.

Таблиця 4.5 – Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1.	Управління інформаційною безпекою	Малий бізнес; Середній бізнес;	Наявність вимог до високої швидкості виявлення аномалій	Можливість гнучкого налаштування; Конфігурація параметрів доступності Конфігурація стратегії відмовостійкості;
2.	Оптимізація ресурсів	Малий бізнес; Середній бізнес;	Різні об’єми та характеристики ресурсів	Швидка ініціалізація та конфігурація ресурсів; Підвищення коефіцієнту використання ресурсів

Основною характеристикою потенційних клієнтів є те, що вони зацікавлені у програмі, яка з більшою точністю зможе виявляти мережеві аномалії. Основними частинами ринку є: великі, середні та малі компанії, які вже намагаються виявити мережеві аномалії.

Далі, необхідно визначити фактори загрози, які наведені у таблиці 4.6.

Таблиця 4.6 – Фактори загроз

№ п/п	Фактори	Зміст загрози	Можлива реакція компанії
1.	Крадіжка інтелектуальної власності	Крадіжка ідеї або ключової інтелектуальної інновації	Відсудження прав інтелектуальної власності; Забезпечення якісного захисту інформації; Зміна методики шифрування приватного ключа; Попередження користувачів із подальшою співпрацею для мінімізації фактор загрози
2.	Отримання несанкціонованого доступу сторонніми особами	Хакерська атака що може призвести до компрометації даних клієнтів	Залучення спеціалістів з інформаційної безпеки; Використання засобів шифрування та резервного копіювання;
3.	Відсутність ринку	Відсутність шляху збуту товару внаслідок помилкового орієнтування	Ретельний розгляд проблем потенційних клієнтів; Залучення експертів та менторів; Консультації зі спеціалістами
4.	Недостача капіталовкладень	Витрачені усі кошти до моменту виходу на ринку	Пошук нових джерел інвестицій

Були розглянуті загрози які можуть виникнути під час реалізації стартап-проекту. Найбільшою загрозою є досить висока конкуренція яка може з'явитися

через появу великої компанії. Стартапу необхідно провести дослідження на предмет використання користувачами програми. Це дозволить збільшити клієнтську базу та покращити розуміння користувачів. Реакція на вихід великої компанії на ринок може бути: вихід стартапу, поглинання великою компанією стартапу, вихід на IPO та залучення інвестицій. Фактори можливостей наведені в таблиці 4.7.

Таблиця 4.7 – Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1.	Отримання інвестицій	Отримання капіталу що необхідний для реалізації продукту	Розробка продукту
2.	Успішна маркетингова політика	В результаті проведеної маркетингової політики отримана висока зацікавленість користувачів	Підтримка стабільної роботи системи та проведення масштабування системи; Збільшення цін на використання сервісу; Використання подібної маркетингової стратегії надалі для залучення нових користувачів
3.	Поглинання конкурентами	Пропозиція купівлі проекту або розроблених технологій одним із конкурентів	Розвиток розроблених технологій; Оцінка вартості розроблених технологій

Були описані основні позитивні фактори: збільшення кількості покупців та недовіри до конкурентів, а також ріст зацікавленості до автоматичного виявлення мережевих аномалій. Реакція компанії полягає у використанні можливостей які можуть спричинити позитивну динаміку сприйняття продукту.

В таблиці 4.8 проведено аналіз конкуренції, який існує на ринку та тип конкуренції: олігополія; за галузевою ознакою: внутрішньогалузева. Також проведено дослідження яка є конкуренція за видами товарів: товарно-видова; інтенсивність: марочна. Також описані дії стартапу необхідні для забезпечення конкурентоспроможності. Дії стартапу на початку; фокусування на вирішенні специфічної проблеми та зниження кількості помилкових спрацювань майже у три рази.

Таблиця 4.8 – Ступеневий аналіз конкуренції ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність, підприємств (можливі дії компанії, щоб бути конкурентоспроможною)
Олігополія	Незначна кількість конкурентів; Велика ринкова сила; Схожість використовуваних технологій	Інформування ринку щодо появи нової платформи управління системою виявлення мережевих аномалій
Галузевий	Загроза появи нових конкурентів; Виркова влада споживачів; Висока потреба у товарі	Інформування ринку щодо якості використовуваної новаторської технології; Пропозиція гнучких цін
Внутрішньогалузева	Діяльність в одній галузі економіки; Надання сервісів одного типу	Зменшення вартості сервісу; Примноження каналів розподілу
Цінова	Використання цін для покращення економічних умов збуту	Зменшення вартості платформи; Використання нових каналів розподілу

Кінець таблиці 4.8

Товарно-видова	Надання різних сервісів одного типу	Маркетингова політика
Марочна	Пропозиція схожої платформи; Спільна цільова аудиторія	Інформування ринку щодо появи нової платформи управління подійними логами системи

Далі в таблиці 4.9 наведено перелік факторів конкурентоспроможності на потенційному ринку який зроблений за допомогою аналізу конкуренції в галуз за методом М. Портера.

Таблиця 4.9 – Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	“McAfee NSP” “Hillstone NIPS” “Huawei NIP”	Розмір капіталовкладень; Забезпечення гнучких цін; Доступ до каналів розподілу; Витрати на масштабах	Відсутні	Змінні витрати: Виробничі непрямі дегресивні - Системи інформації: пропаганда, реклама та директ-маркетинг,-Рівень	Копіювання функціоналу; Монополізація дистриб’юторів; Демпінгування

Продовження таблиці 4.9

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
				чутливості до цін: споживачі орієнтовані на цінність продукту- Продуктова диференціація: якість, спосіб отримання сервісу,швидкість обслуговування Методи контролю якості: тестування та профілювання, прототипування, інспектування коду, аналіз архітектури програмного забезпечення	
Висновки	CR4 = 92%; Індекс Херфіндаля-Хіршмана (HHI) = 6565; Значення показників	Можливості входу на ринок забезпечує мінімізацію цін,	Відсутні	Клієнти диктують умови гнучкості цінової політики, високої і довгострокової	Пропонування вигідних умов дистриб’юторам, забезпечення захисту інтелектуальної власності,

Кінець таблиці 4.9

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	вказує на високу концентрацію (монополізацію) даного ринку	швидкість та простота надавання послуги споживачам і співпраця із головними гравцями ринку. В результаті аналізу проектів на народно-громадських інтернет-платформах потенційних конкурентів знайдено не було		якості послуг та наявність кооперації із сервісами, що вони використовують	гнучкість цінової політики

Конкурентна ситуація на ринку є дуже привабливою через те що конкурентні програми мають досить велику кількість помилкових спрацювань. Створювана програма знижує кількість помилкових спрацювань майже у три

рази. Основними сильними сторонами продукту покращений алгоритм виявлення, надійність та швидкість. Обгрунтування основних факторів конкурентоспроможності наведено в таблиці 4.10.

Таблиця 4.10 – Обгрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1.	Унікальність сервісу	Розроблений продукт має унікальне співвідношення ціна / якість для свого цінового діапазону
2.	Цінова політика	Отримання прибутку здійснюється за рахунок гнучкої моделі оплати
3.	Модель “бізнес для бізнесу”	Бізнес модель грунтується на співпраці із іншими платформами управління інфраструктурами. Даний підхід дозволяє обійти цінову конкуренцію на ринку цільової аудиторії

Проаналізувавши можливості роботи на ринку з огляду на конкурентну ситуацію можна зробити висновок: оскільки з огляду на конкурентну ситуацію можна зробити висновок: оскільки кожний з існуючих продуктів не впливає у великій мірі на поточну ситуацію на ринку в цілому, кожний з існуючих продуктів має свою специфічну сферу використання та свої позитивні та негативні сторони щодо рішення певних типів задач, то робота та вихід на даний ринок є можливою і реалізованою задачею.

Для виходу на ринок продукт повинен мати функціонал зо відсутній у продуктів-аналогів, повинен задовольняти потреби користувачів, мати необхідний та достатній функціонал з конфігуруванням, підтримку зі сторони розробників та можливість розробки спеціального функціоналу за відповідною

ліцензією. Порівняльний аналіз сильних та слабких сторін наведено в таблиці 4.11.

Таблиця 4.11 – Порівняльний аналіз сильних та слабких сторін “McAfee NS”

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з “McAfee NS”						
			-3	-2	-1	0	1	2	3
1.	Унікальність сервісу	15						+	
2.	Цінова політика	17							+
3.	Модель “бізнес для бізнесу”	13					+		

Були визначені сильні та слабкі сторони продукту в порівнянні з рішеннями конкурентів. Сильною стороною є використання покращеного методу виявлення мережевих аномалій. В таблиці 4.12 наведений SWOT-аналіз стартап-проекту.

Таблиця 4.12 – SWOT-аналіз стартап-проекту

Сильні сторони: Якість та довготривалість; Низькі ціни;	Слабкі сторони: Нестача капіталовкладень; Бізнес-модель залежить від політики окремих бізнесів;
Можливість: Інвестиції; Реалізація бізнес-моделі; Розширений функціонал; Висока зацікавленість цільової аудиторії;	Загрози: Крадіжка інтелектуальної властивості; Компрометація даних клієнтів; Відсутність ринку;

У стартапу є наступні сильні та слабкі сторони. Використання покращеного методу виявлення мережевих аномалій, краща швидкість та надійність. Слабкими сторонами є - недостатні капіталовкладення та залежність бізнес-

моделі від політики окремих бізнесів. Ринкові можливості стартапу мають наступні можливості та загрози: можливості - інвестиції, висока зацікавленість цільової аудиторії, розширений функціонал; загрози - крадіжка інтелектуальної властивості, компрометація даних клієнтів.

На базі вже проведеного SWOT - аналізу було розроблені альтернативні можливості ринкового впровадження, які наведенеі нижче в таблиці 4.13.

Таблиця 4.13 – Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтований комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1.	Розробка власних засобів віртуалізації	Ймовірне	12 місяців
2.	Маркетингова кампанія для приваблювання користувачів	Малоймовірне	2 місяці
3.	Пропонування безкоштовних тарифів	Малоймовірне	1 місяць
4.	Пошук бізнесів інших галузей для співпраці	Дуже ймовірне	6 місяців
Обрана альтернатива: Пошук бізнесів з інших галузей для співпраці			

Альтернативою ринкового впровадження є пошук бізнес партнерів із інших галузей.

4.4. Розроблення ринкової стратегії проекту

В цьому розділі було сформовано стратегію охоплення ринку а точніше груп потенційних кінцевих користувачів. В таблиці 4.14 наведено вибір цільових груп потенційних споживачів.

Таблиця 4.14 – Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтований попит в межах цільової групи(сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1.	Адміністратори інфраструктур безпеки малого бізнесу	Висока	65%	Середня	Низькі бар‘єри входу
2.	IT-підрозділи середнього бізнесу	Висока	78%	Середня	Низькі бар‘єри входу
3.	Власники систем виявлення вторгнень	Мала	35%	Середня	Високі бар‘єри входу
Які цільові групи обрано: адміністратори інфраструктури безпеки, малого IT-підрозділи середнього бізнесу					

Відповідно до проведеного аналізу можна зробити висновок, що підходящою цільовою групою для розповсюдження даного програмного продукту є працівники інфраструктури безпеки малого бізнесу, IT-підрозділи середнього бізнесу в цілому та будь-які підприємства котрі використовують схеми IDS. Відповідно до стратегії охоплення ринку збуту товару обрано стратегію масового маркетингу, оскільки для підприємств, IT працівників та IT

компаній у цілому надається стандартизований продукт з можливістю розширення функціональності за домовленістю (відповідно до ліцензії). В таблиці 4.15 наведено визначення базової стратегії розвитку.

Таблиця 4.15 – Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1.	Надання платформи малому та середньому бізнесу	Вибірковий розподіл	Здатність протистояти прямим конкурентам; Низькі витрати; Ефективна співпраця;	Стратегія диференціації

Стратегія розвитку продукту є диференціація, яка має на меті позиціювання. Через те що основними перевагами продукту є надійність та швидкість.

В таблиці 4.16 наведено визначення базової стратегії конкурентної поведінки.

Таблиця 4.16 – Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект “першопрохідцем” на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
1.	Ні	Забирати та залучати нових	Веб-інтерфейс керування інфраструктурою; Інтелектуальних розподіл	Стратегія лідера; Розширення первинного попиту.

Кінець таблиці 4.16

№ п/п	Чи є проект “першопрохідцем” на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
			обчислювальних ресурсів	

Було обрано стратегію конкурентної поведінки, яка полягає у агрессивному маркетингу. Також було визначено, що компанія буде копіювати вже добре відомі інтерфейси та більш надійний та швидкий продукт.

Визначення стратегії позиціювання показано в таблиці 4.17.

Таблиця 4.17 – Визначення стратегії позиціювання

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможності позиції власного стартап-проекту	Вибір асоціації, які мають сформувати комплексну позицію власного проекту (три ключових)
1.	Відповідальність затвердженим характеристикам; Високий ступінь надійності системи; Простий інтерфейс адміністратора;	Стратегія диференціації	Формування регулярного попиту; Збільшення разового використання послуги; Виявлення нових груп послуги; Виявлення нових груп споживачів;	Інноваційність технології; Низькі ціни; Простота використання

Кінець таблиці 4.17

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможності позиції власного стартап-проекту	Вибір асоціації, які мають сформувати комплексну позицію власного проекту (три ключових)
	Гнучка цінова політика; Оперативна підтримка продукту		Нові напрямки застосування існуючої послуги	

Відповідно до проведеного аналізу можна зробити висновок, що стартап-компанія вибирає як базову стратегію розвитку - стратегію диференціації, як базову стратегію конкурентної поведінки - стратегію заняття конкурентної ніші.

4.5 Розробка маркетингової програми стартап-проекту

Для розробки маркетингової стратегії стартапу, необхідно сформувати маркетингову концепцію продукту. В таблиці 4.18 наведено ключові переваги концепції товару.

Таблиця 4.18 – Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1.	Управління інформаційною безпекою	Реалізація відмовостійкості та високодоступності застосунків	Якість надання послуг; Інноваційність технологій що використовується Простота використання; Цінова перевага;
2.	Управління обчислювальними ресурсами	Швидка ініціалізація та конфігурація ресурсів; Підвищення використання ресурсів	Якість надання послуг; Інноваційність технологій що використовується Цінова перевага

Далі потрібно створити трирівневої маркетингової моделі товару яка представлена у таблиці 4.19.

Таблиця 4.19 – Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом.	Програмний продукт що надає можливість пришвидшити виявлення мережевих аномалій		
II. Товар у реальному виконанні	Властивість/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	Кількість		1 шт.
	Якість: стандарти якості постачання програмних продуктів		
	Пакування: комп’ютерний диск/дискета/флешка		

Кінець таблиці 4.19

	Марка: Kharkhonov IDS Platform
III. Товар із підкріпленням	Програмний продукт
	Програмний продукт, технічна підтримка та підписка на оновлення
За рахунок чого потенційний товар буде захищено від копіювання: захист інтелектуальної власності	

У таблиці вище описані три рівні моделі проекту. Стартап-проект буде виконаний у вигляді програми та допомагатиме автоматично виявляти мережеві аномалії. Основними характеристиками є: швидкість та надійність. Програму буде захищено від копіювання патентом.

Далі, було визначено цінову межу за допомогою експертного методу, яка наведена в таблиці 4.20.

Таблиця 4.20 – Визначення меж встановлення межі

№ п/п	Рівень цін на товари-аналоги	Рівень цін на товари аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1.	50 тис. грн. - 100 тис. грн.	50 тис. грн. - 200 тис. грн.	150 тис. грн. / міс.	40 тис. грн. - 60 тис. грн.

Було визначено межі можливих цін продажу програми; рівень цін товарів замінників та доходів споживачів. Верхньою межею є 40 тис. грн, нижньою 60 тис. грн. відповідно.

Оптимальна система збуту у межах якої можуть прийматися потенційні рішення наведена у таблиці 4.21.

Таблиця 4.21 – Формування системи збуту

Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
Закупівля здійснюється через довірені джерела	Інформація користувачів; Доступ користування сервісом	Канал одного рівня	Селективна з використовуванням комбінованого каналу збуту

Було розроблено систему збуту, яка має форму продажу. Збут буде виконуватись за власними каналами.

Далі було розроблено концепцію маркетингових комунікацій стартапу, які наведені у таблиці 4.22.

Таблиця 4.22 – Концепція маркетингових комунікацій

Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користується цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
Автоматизація бізнес процесів; Вимоги до високої доступності та відмовостійкості	Інтернет	Послідовність в реалізації обраної позиції; Доступність та об’єктивність інформації про фірму та товар; Унікальність послуги	Формування у цільової аудиторії обізнаності про появу нового продукту; Інформування користувачів про властивості та переваги продукту; Інформування користувачів	Раціоналістична стратегія реклами.

Кінець таблиці 4.22

Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користується цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
			про нові способи використання відомого продукту; Пояснення цільовій аудиторії принципу роботи платформи; Виправити у користувачів неправильні представлення про продукт	

Після аналізу маркетингової кампанії, було створено концепцію потенційних маркетингових комунікацій. Завантаження для кожної з платформ буде можливе з офіційного сайту, а каналом комунікацій є інтернет. Рекламне повідомлення повинно мати форму демонстраційного відео та показувати швидкість виявлення мережевих аномалій.

Висновок до розділу

Як результат, було створено ринкову (маркетингову) програму, що включає в себе визначення ключових переваг концепції потенційного товару,

опис моделі товару, визначення меж встановлення ціни, формування системи збуту та концепцію маркетингових комунікацій.

За допомогою зробленого аналізу, можна припустити, що існує можливість ринкової комерціалізації проекту через те, що на ринку систем виявлення вторгнень на базі аналізу аномалій все ще наявний попит на такі системи, до того ж рентабельність розробки таких систем є досить високою.

ВИСНОВОК

У даній магістерській дисертації було описано задачу виявлення мережевих аномалій. Було описано актуальність задачі та проаналізовано застосування систем виявлень вторгнень для запобігання атак, так як саме ці системи являються основними складовими в забезпеченні безпеки IT інфраструктури. Головна загроза для таких інфраструктур це кібератаки, які постійно розвиваються та яких з кожним днем стає все більше і більше. Що запобігти таким атак, системи виявлення вторгнень аналізують вхідний та вихідний трафік, в результаті чого можна вжити попереджувальні заходи для захисту мережі.

Після огляду задачі виявлення аномалій, було досліджено основні методи вирішення постанвленої задачі, а саме: статистичні моделі, класифікація, кластеризація, бази знань, м’які обчислення та комбіноване навчання. Був проведений порівняльний аналіз цих методів, в результаті якого було відібрано декілька найбільш підходящих для поставленої задачі методів.

Далі, після повного аналізу задачі та наявних засобів до вирішення цієї задачі, ціллю було розробити систему виявлення мережевих аномалій за допомогою алгоритмів машинного навчання. Для проекту було обрано датасет CICIDS2017 через його актуальність, широко різноманіття атак та різних мережевих протоколів (Поштовий протокол, SSH, FTP, HTTP та HTTPS). Цей датасет містить більше ніж 80 різних ознак що визначають мережевий потік. Під час аналізу та розробки було пораховано ваги значущості ознак за допомогою алгоритму Random Forest Regressor для відбору найбільш значущих ознак для використання їх в алгоритмах машинного навчання. Для розрахунку ваг ознак використовувалось два підходи: в першому ваги розраховувались для кожної

атаки окремо, в другому підході розрахунок проводився на всьому наборі даних. В результаті було відібрано загальні ознаки, які важливі для всіх атак. Далі, було застосовано кілька популярних алгоритмів машинного навчання до отриманих даних та з використанням відібраних ознак. Отримані результати та значення критерію F-measure виглядають наступним чином (F-measure приймає значення між 0 та 1): Naive Bayes: 0.86, QDA: 0.86, Random Forest: 0.94, ID3: 0.95, K Nearest Neighbours: 0.94, MLP: 0.83, and XGBoost: 0.97.

В останньому розділі описано стратегії та підходи з розробленням стартап-проектуу, визначено наявінсть попиту, динаміки та рентабельності роботи на ринку. Було вказано що існує можливість ринкової комерціалізації проекту. Розглянувши потенційні групи клієнтів, бар'єри входження, стан конкуренції та конкурентоспроможності проекту, було встановлено що проект є перспективним. Було розглянуто та вибрано альтернативи впровадження стартап проекту та доведено доцільність імплементації проекту.

ПЕРЕЛІК ПОСИЛАНЬ

1. Heady R., Luger G., Maccabe A. and Servilla M. The Architecture of a Network Level Intrusion Detection System. *Computer Science Department*, University of New Mexico, Tech. Rep. TR-90. 1990. 21 p. URL: https://www.osti.gov/servlets/purl/425295 (дата звернення: 02.09.2020)

2. Anderson J. P. Computer Security Threat Monitoring and Surveillance. James P Anderson Co, Fort Washington, Pennsylvania, Tech. Rep., 1980. 56 p. URL: https://csrc.nist.gov/csrc/media/publications/conference-paper/1998/10/08/proceedings-of-the-21st-nissc-1998/documents/early-cs-papers/ande80.pdf (дата звернення: 03.09.2020)

3. Chandola V., Banerjee A., Kumar V. Anomaly Detection: A Survey. *ACM Computing Surveys*. 2009. Vol. 41, no. 3. P. 15–38.

4. Ghorbani A., Lu W., Tavallaee M. Network Intrusion Detection and Prevention: Concepts and Techniques. New York: Springer-verlag, 2009. 216 p.

5. Ning P., Jajodia S. Intrusion Detection Techniques. HBidgoli (Ed.), The Internet Encyclopedia, 2003. URL: https://doi.org/10.1002/047148296X.tie097 (дата звернення: 06.09.2020)

6. Daniel B., Julia C., Sushil J, Ningning W. ADAM: a test bed for exploring the use of data mining in intrusion detection. *ACM SIGMOD Record*. 2001. Vol. 30, no. 4, pp. 15–24.

7. Chan P. K., Mahoney M. V., Arshad M. H. A machine learning approach to anomaly detection. Department of Computer Science, Florida Institute of Technology, Tech. Rep. CS-2003-06, 2003. URL: https://cs.fit.edu/media/TechnicalReports/cs-2003-06.pdf (дата звернення: 06.09.2020)

8. Bhuyan M. H., Bhattacharyya D. K., Kalita J. K. Surveying Port Scans and Their Detection Methodologies. *The Computer Journal*. 2011. Vol. 54, no. 10. P. 1565–1581.

9. Thottan M., Ji C. Anomaly detection in IP networks. *IEEE Transactions on Signal Processing*. 2003. Vol. 51, no. 8. PP. 2191–2204.

10. Park C., Won Y. J., Kim M. S., Hong J. W. Towards automated application signature generation for traffic identification. *Proc. of the IEEE/IFIP Network Operations and Management Symposium: Pervasive Management for Ubioquitous Networks and Services*. 2008. No. 10. P. 160–167.

11. Kumar V. Parallel and distributed computing for cybersecurity. *IEEE Distributed Systems Online*. 2005. Vol. 6, no. 10. 9 p.

12. Kayacik H. G., Zincir-Heywood A. N., Heywood M. I. Selecting Features for Intrusion Detection: A Feature Relevance Analysis on KDD 99 Intrusion Detection Datasets. *Proc. of the 3rd Annual Conference on Privacy, Security and Trust*. 2005. URL: https://web.cs.dal.ca/~kayacik/papers/PST05.pdf (дата звернення: 10.09.2020)

13. Tan P. N., Steinbach M., Kumar V. Introduction to Data Mining. Boston: Addison-Wesley, 2005. 166 p.

14. Lesot M. J., Rifqi M. Anomaly-based network intrusion detection: Techniques, systems and challenges. *International Journal of Knowledge Engineering and Soft Data Paradigms*. 2009. Vol. 1, no. 1. P. 63–84.

15. Cha S. H. Comprehensive Survey on Distance/Similarity Measures between Probability Density Functions. *International Journal of Mathematical Models and Methods in Ap-plied Science*. 2007. Vol. 1, no. 4. P. 300–307.

16. Choi S., Cha S., Tappert C. C. A Survey of Binary Similarity and Distance Measures. *Journal of Systemics, Cybernetics and Informatics*. 2010. Vol. 8, no. 1. P. 43–48.

17. Lesot M. J., Rifqi M., Benhadda H. Similarity measures for binary and numerical data: a survey. *International Journa lof Knowledge Engineering and Soft Data Paradigms*. 2009. Vol. 1, no. 1. P. 63–84.

18. Boriah S., Chandola V., Kumar V. Similarity measures for categorical data: A comparative evaluation. *SDM 2008*: Proc. of the 8th SIAM International Conference on Data Mining, Atlanta, Georgia, USA, 24 April - 26 April, 2008. Minnesota: Spring, 2008. P .243–254.

19. Gan G., Ma C., Wu J. Data Clustering Theory, Algorithmsand Applications. SIAM, 2007. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1751-5823.2007.00039_2.x (дата звернення: 13.10.2020)

20. Hsu C., Wang S. H. An integrated framework for visualized and exploratory pattern discovery in mixed data. *IEEE Transactions on Knowledge and Data Engineering*. 2005. Vol. 18, no. 2. P. 161–173.

21. Joshi M. V., Agarwal R. C., Kumar V. Mining needle in a haystack: classifying rare classes via two-phase rule induction. *SIGMOD 2001*: Proc. of the 7th ACM SIGKDD Internationa lConference on Knowledge Discovery and Data Mining, Santa Barbara, California, USA, 21 May – 25 May, 2001. NY: ACM. P. 293–298.

22. Theiler J., Cai D. M. Resampling approach for anomaly detection in multi spectral images. *Proceedings of SPIE*. 2003. Vol. 5093. P. 230–240.

23. Fujimaki R., Yairi T., Machida K. An approach to spacecraft anomaly detection problem using kernel feature space. *SIGKDD 2005*: Proc. of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, Chicago, Illinois, USA, 15 August – 25 August, 2005. NY: ACM. P. 401–410.

24. Portnoy L., Eskin E., Stolfo S. J. Intrusion detection with unlabeled data using clustering. *DMSEC 2001*: Proc. of The ACM Workshop on Data Mining Applied

to Security, Waschington DC, USA, 3 October – 15 October, 2001. NY: ACM. P. 354-367.

25. Nguyen H. H., Harbi N., Darmont J. An efficient local region and clustering-based ensemble system for intrusion detection. IDEAS 2001: Proc. of the 15th Symposium on Internationa lDatabase Engineering & Applications, Lisboa, Portugal, 3 September – 10 September, 2001. NY: ACM. P. 185–191.

26. Dash M., Liu H. Feature Selection for Classification. *Intelligent Data Analysis*. 1997. Vol. 1. P. 131–156.

27. Chen Y., Li Y., Cheng X. Q., Guo L. Survey and taxonomy of feature selection algorithms in intrusion detection system. Heidelberg: Springer-Verlag, 2006. 356 p.

28. Li Y., Wang J. L., Tian Z., Lu T., Young C. Building lightweight intrusion detection system using wrapper-based feature selection mechanisms. *Computers & Security*. 2009. Vol. 28, no. 6. P. 466–475.

29. Nguyen H. T., Franke K., Petrovic S. Towards a Generic Feature-Selection Measure for Intrusion Detection. *ICPR 2010*: Proc.of the 20th International Conference on Pattern Recognition, Istanbul, Turkey, 14 August – 21 August, 2010. Waschington: IEEE. P. 1529–1532.

30. Sung H., Mukkamala S. Identifying Important Features for Intrusion Detection Using Support Vector Machinesand Neural Networks. *SAINT 2003:* Proc. of the Symposium on Applications and the Internet, Orlando, Florida, USA, 27 January – 31 January, 2003. Waschington: IEEE CS. P. 209–217.

31. Peng H., Long F., Ding C. Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2005. Vol. 27, no. 8. P. 1226–1238.

32. Amiri F., Yousefi M. M. R., Lucas C., Shakery A., Yazdani N. Mutual information-based feature selection for intrusion detection systems. *Journal of Network and Computer Applications*. 2001. Vol. 34, no. 4. P. 1184–1199.

33. Anscombe J., Guttman I. Rejection of outliers. *Technometrics*. 1960. Vol. 2, no. 2. P. 123–147.

34. Eskin E. Anomaly detection over noisy data using learned probability distributions. *ICML 2000:* Proc. of the 7th International Conference on Machine Learning, Stanford, CA, USA, 29 June – 2 July, 200. Heidelberg: Springer-Verlag. P. 255–262.

35. Desforges M., Jacob P., Cooper J. Applications of probability density estimation to the detection of abnormal conditions in engineering. *Institute of Mechanical Engineers*. 1998. Vol. 212, no. 2. P. 687–703.

36. Manikopoulos C., Papavassiliou S. Network Intrusion and Fault Detection: A Statistical Anomaly Approach. *IEEE Communications Magazine*. 2002. Vol. 40, no. 10. P. 76–82.

37. Mahoney M. V., Chan P. K. Learning rules for anomaly detection of hostile network traffic. *ICDM* 2003: Proc. of the 3rd IEEEInternational Conference on Data Mining, Melbourne, FL, USA, 22-22 November, 2003. Washington: IEEECS. P. 145-176.

38. Wang K., Stolfo S. J. Anomalous Payload-Based Network Intrusion Detection. *RAID 2004*: Proc. of the Recent Advances in Intrusion Detection, Sophia Antipolis, France, 15 September – 17 September, 2004. Waschington: Springer. P. 203–222.

39. Sundaram A. An introduction to intrusion detection. *Cross-roads*. 1996. Vol. 2, no. 4. P. 3–7.

40. Song X., Wu M., Jermaine C., Ranka S. Conditional Anomaly Detection. *IEEE Transactions on Knowledge and Data Engineering*. 2007. Vol. 19. P. 631–645.

41. Chhabra P., Scott C., Kolaczyk E. D., Crovella M. Distributed Spatial Anomaly Detection. *IEEE INFOCOM 2008:* Proceedings ofthe 27th IEEE International Conference on Computer Communications, Phoenix, AZ, USA, 13 April – 18 April, 2008. NY: IEEE. P. 1705–1713.

42. Lu W., Ghorbani A. A. Network Anomaly Detection Based on Wavelet Analysis. *EURASIP Journal on Advancesin Signal Processing*. 2009. Vol. 2009, no. 837601. P. 43 – 67.

43. Wattenberg S., Perez J. I. A., Higuera P. C., Fernandez M. M., Dimitriadis I. A. Anomaly Detection in Network Traffic Based on Statistical Inference and Stable Modeling. *IEEE Transactions on Dependable and Secure Computing*. 2001. Vol. 8, no. 4. P. 494–509.

44. Zhang Z., Li J., Manikopoulos C. N., Jorgenson J., Ucles J. HIDE: a Hierarchical Network Intrusion Detection System Using Statistical Preprocessing and Neural Network Classification. *IEEE 2001*: in Proc. of IEEE Man Systems and Cybernetics Information Assurance Workshop, West Point, NY, USA, 5 Hune – 6 June, 2001. Waschington: Springer. P. 56 – 99.

45. Patcha A., Park J. M. An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer Networks*. 2007. Vol. 51, no. 12. P. 3448–3470.

46. Garcia-Teodoro P., Diaz-Verdejo J., Macia-Fernandez G., Vazquez E. Anomaly-based network intrusion detection: Techniques, systems and challenges. *Computers & Security*. 2009. Vol. 28, no. 1-2. P. 18–28.

47. B Daniel, C. Julia, J. Sushil, and W. Ningning, “ADAM: a test bed for exploring the use of data mining in intrusion detection”*ACM SIGMOD Record*, vol. 30, no. 4, pp. 15–24, 2001.

48. L. Ertoz, E. Eilertson, A. Lazarevic, P. Tan, V. Kumar, and J. Srivastava. Data Mining - Next Generation Challenges and Future Directions. Massachusetts: AAAI Press, 2004. 528 p.

ДОДАТОК А ЛІСТИНГ ПРОГРАМИ

```
import pandas as pd
import numpy as np
import glob
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.offline as py
import plotly
import plotly.graph_objs as go
import plotly.figure_factory as ff
plotly.io.renderers.default = 'colab'
from plotly.subplots import make_subplots
df.columns = df.columns.str.lstrip()
df.info()
df.drop(["Source IP", "Destination IP", "Timestamp", "Flow ID"], inplace=True, axis=1)
df.head()
import pandas as pd
import numpy as np
import glob
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.offline as py
import plotly
import plotly.graph_objs as go
```

```
import plotly.figure_factory as ff
plotly.io.renderers.default = 'colab'
from plotly.subplots import make_subplots
df.columns = df.columns.str.lstrip()
df.info()
df.drop(["Source IP", "Destination IP", "Timestamp", "Flow ID"], inplace=True, axis=1)
df.head()
print(f"Missing values: {df.isnull().sum().sum()}")
df.replace([np.inf, -np.inf], np.nan, inplace=True)
print(f"Missing values: {df.isnull().sum().sum()}")
df.dropna(inplace=True)
df.shape
df["attack"] = df["Label"].apply(lambda x: 0 if x=="BENIGN" else 1)
attacks = df[df["attack"]==1]
benign = df[df["attack"]==0]
print(f"Attack records: {len(attacks)}\nBenign records: {len(benign)}")
benign = benign.sample(frac=0.3).reset_index(drop=True)
print(f"Attack records: {len(attacks)}\nBenign records: {len(benign)}")
df = pd.concat([attacks, benign])
sns.countplot(x="attack", data=df)
plt.xticks([0,1], ["Normal", "Attack"])
plt.title("CICIDS2017 distribution after subsetting")
plt.xlabel("")
plt.show()
 from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(df,
```

```
                                        df["attack"],
                                        test_size=0.50,
                                        random_state=0,
                                        stratify=df["Label"],
                                        shuffle=True)
f, axes = plt.subplots(2, 2, figsize=(12, 10))
sns.countplot(x="attack", data=X_train, ax=axes[0,0])
sns.countplot(x="attack", data=X_test, ax=axes[0,1])
sns.countplot(x="Label", data=X_train, ax=axes[1,0], order =
X_train['Label'].value_counts().index)
sns.countplot(x="Label", data=X_test, ax=axes[1,1], order =
X_test['Label'].value_counts().index)
axes[0,0].set_title("Training data distribution")
axes[1,0].set_title("Training data distribution")
axes[0,1].set_title("Testing data distribution")
axes[1,1].set_title("Testing data distribution")
axes[1,0].tick_params('x', labelrotation=90)
axes[1,1].tick_params('x', labelrotation=90)
axes[0,0].set_xticklabels(["Normal", "Attack"])
axes[0,1].set_xticklabels(["Normal", "Attack"])
axes[0,0].set_xlabel("")
axes[0,1].set_xlabel("")
axes[1,0].set_xlabel("")
axes[1,1].set_xlabel("")
plt.subplots_adjust(wspace=0.25)
train = X_train.copy()
test = X_test.copy()
```

```
train.loc[:,"label"] = y_train
test.loc[:,"label"] = y_test

train.to_csv("drive/My Drive/Colab Notebooks/data/train.csv", header=True, index=False)
test.to_csv("drive/My Drive/Colab Notebooks/data/test.csv", header=True, index=False)
train.label.unique()
train = pd.read_csv("drive/My Drive/Colab Notebooks/data/train.csv")
test = pd.read_csv("drive/My Drive/Colab Notebooks/data/test.csv")
import numpy as np
import pandas as pd
import tensorflow as tf
import tensorflow_probability as tfp
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import average_precision_score, f1_score
from sklearn.metrics import recall_score, accuracy_score
train_data = pd.read_csv("drive/My Drive/Colab Notebooks/data/train.csv")
test_data =  pd.read_csv("drive/My Drive/Colab Notebooks/data/test.csv")
important_features_origin = ['Bwd Packet Length Std', 'Source Port', 'Destination Port','Flow IAT Mean','Fwd Packet Length Std','Flow IAT Std','Total Length of Fwd Packets', 'Total Fwd Packets','Fwd Packet Length Min','Bwd Packet Length Mean','Total Backward Packets','Total Length of Bwd Packets','Flow Duration','Flow Packets/s','Flow Bytes/s','Bwd Packet Length Max','Fwd Packet Length Max','Fwd Packet Length Mean','Bwd Packet Length Min','Protocol']
```

```
important_features = ['Bwd Packet Length Std', 'Source Port', 'Destination Port','Flow IAT Mean','Fwd Packet Length Std','Flow IAT Std','Total Length of Fwd Packets', 'Total Fwd Packets','Fwd Packet Length Min','Flow Duration','Flow Packets/s','Flow Bytes/s','Fwd Packet Length Mean','Bwd Packet Length Min','Protocol']
y_train = np.array(train_data["label"])
train_data.drop(['label'], inplace=True, axis=1)
train_data = train_data[important_features]
y_test = np.array(test_data["label"])
test_data.drop(['label'], inplace=True, axis=1)
test_data = test_data[important_features]
scaler = StandardScaler()
train_data[train_data.columns] = scaler.fit_transform(train_data[train_data.columns])
test_data[test_data.columns] = scaler.transform(test_data[test_data.columns])
print(f"Train data shape: {train_data.shape} Train label shape: {y_train.shape}")
print(f"Test data shape: {test_data.shape} Test label shape: {y_test.shape}")
FEATURES = train_data.shape[1]
train_length = train_data.shape[0]
test_length = test_data.shape[0]
labels = df[df[' Label'] != 'BENIGN'][' Label'].value_counts()
df_labels = pd.DataFrame({'labels': labels.index,
                          'values': labels.values
                         })
iplot([
    go.Pie(labels=labels.index, values=labels.values)
])
plt.figure(figsize = (15,10))
sns.heatmap(train_data.corr(), annot=True)
```

```
from sklearn.metrics import jaccard_score
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, accuracy_score, classification_report, cohen_kappa_score
from sklearn.metrics import roc_auc_score, roc_curve,scorer, f1_score, precision_score, recall_score
from sklearn.metrics import precision_recall_curve, average_precision_score
import statsmodels.api as sm
from yellowbrick.classifier import DiscriminationThreshold
def churn_prediction(algorithm, training_x, testing_x, training_y, testing_y, cf):
    algorithm.fit(training_x, training_y)
    predictions = algorithm.predict(testing_x)
    probabilities = algorithm.predict_proba(testing_x)
    print('Algorithm:', type(algorithm).__name__)
    print("Accuracy Score:", accuracy_score(testing_y, predictions))
    conf_matrix = confusion_matrix(testing_y, predictions)
    model_roc_auc = roc_auc_score(testing_y, predictions)
    print("Area under curve:", model_roc_auc)
    print('Jaccard index:', jaccard_score(testing_y, predictions),"\n")
    sns.heatmap(conf_matrix, annot=True,fmt='g')
    if cf in ['coefficients', 'features']:
        if cf == 'coefficients':
            coefficients = pd.DataFrame(algorithm.coef_.ravel())
        elif cf == 'features':
            coefficients = pd.DataFrame(algorithm.feature_importances_)
        column_df = pd.DataFrame(training_x.columns.tolist())
        coef_sumry = (pd.merge(coefficients, column_df, left_index=True,
```

```
                    right_index=True, how="left"))
    coef_sumry.columns = ["coefficients", "features"]
    coef_sumry = coef_sumry.sort_values(by = "coefficients", ascending=False)
    plt.figure(figsize=(10,5))
    chart = sns.barplot(x=coef_sumry['features'], y=coef_sumry['coefficients'])
    chart.set_xticklabels(chart.get_xticklabels(), rotation=45,
horizontalalignment='right')
    plt.show()
from sklearn.linear_model import LogisticRegression
logit = LogisticRegression(C=100.0, class_weight=None, dual=False,
fit_intercept=True,
                intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
                penalty='l2', random_state=None, solver='liblinear', tol=0.0001,
                verbose=0, warm_start=False)
churn_prediction(logit, train_data, test_data, y_train, y_test, "coefficients")
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                metric_params=None, n_jobs=1, n_neighbors=4, p=2,
                weights='distance')
churn_prediction(knn, train_data, test_data, y_train, y_test, 'None')
from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier(n_estimators = 500, random_state = 123,
                 max_depth = 15, criterion = "gini", min_samples_leaf=1,
min_samples_split=5)
churn_prediction(rfc, train_data, test_data, y_train, y_test, 'features')
from xgboost import XGBClassifier
xgc = XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
```

```
            colsample_bytree=1, gamma=0, learning_rate=0.9, max_delta_step=0,
            max_depth=12, min_child_weight=1, missing=None, n_estimators=110,
            n_jobs=1, nthread=None, objective='binary:logistic', random_state=0,
            reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=None,
            silent=True, subsample=1)
churn_prediction(xgc, train_data, test_data, y_train, y_test, "features")

from sklearn.neural_network import MLPClassifier
mlp = MLPClassifier(alpha=0.1,hidden_layer_sizes=10, max_iter=2000,
random_state=0, solver='lbfgs')
churn_prediction(mlp, train_data, test_data, y_train, y_test, "None")
cols = [i for i in train_data.columns]
models = {
        'Random Forest': [rfc, cols],
        'XGBoost Classifier': [xgc, cols],
        'Logistic': [logit, cols],
        'KNN Classifier': [knn, cols],
        'MLP Classifier': [mlp, cols],
       }
def model_report(model, training_x, testing_x, training_y, testing_y, name):
    model = model.fit(training_x, training_y)
    predictions = model.predict(testing_x)
    accuracy = accuracy_score(testing_y, predictions)
    recallscore = recall_score(testing_y, predictions)
    precision = precision_score(testing_y, predictions)
    roc_auc = roc_auc_score(testing_y, predictions)
    f1score = f1_score(testing_y, predictions)
```

```
    jaccard_index = jaccard_score(testing_y, predictions)
    kappa_metric = cohen_kappa_score(testing_y, predictions)
    df = pd.DataFrame({"Model"          : [name],
                       "Accuracy"       : [accuracy],
                       "Recall"         : [recallscore],
                       "Precision"      : [precision],
                       "f1-score"       : [f1score],
                       "Roc_auc"        : [roc_auc],
                       "Jaccard_index"  : [jaccard_index],
                       "Kappa_metric"   : [kappa_metric]
                      })
    return df

model_performances_train = pd.DataFrame()
for name in models:
  model_performances_train                                                    =
model_performances_train.append(model_report(models[name][0],
train_data[models[name][1]],
                                                    test_data[models[name][1]],
                                                    y_train,          y_test,          name),
ignore_index=True)
table_train = ff.create_table(np.round(model_performances_train, 4))
py.iplot(table_train)
```